

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ

(повна назва інституту/факультету)


КОНСТРУЮВАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНОЇ АПАРАТУРИ

(повна назва кафедри)

«На правах рукопису»
УДК: 004.932.4

«До захисту допущено»

Завідувач кафедри

 О.М.Лисенко
(підпис) (ініціали, прізвище)

“18” травня 2021 р.

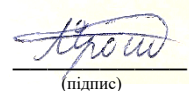
Магістерська дисертація

зі спеціальності (спеціалізації) 172 – Телекомунікації та радіотехніка
(код і назва)

на тему: Методи підвищення роздільної здатності інфрачервоних зображень за допомогою штучних згорткових нейронних мереж

Виконав: студент II курсу, групи ДК-91мн
(шифр групи)

Ярошенко Максим Олександрович
(прізвище, ім'я, по батькові)


(підпис)

Науковий керівник: доц. каф. КЕОА, к.т.н., доц. Варфоломєєв А.Ю.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

Варф
(підпис)

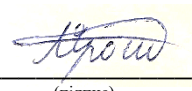
Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент: доц. каф. АМЕС, к.т.н., доц. Попович П.В.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)


(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент 
(підпис)

Київ – 2021 року

**Національний технічний університет України
“Київський політехнічний інститут
імені Ігоря Сікорського”**

Інститут/факультет електроніки
(повна назва)

Кафедра конструювання електронно-обчислювальної апаратури
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо - науковою програмою
Спеціальність 172 – Телекомунікації та радіотехніка
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри


(підпис)

Лисенко О.М.
(прізвище та ініціали)

« 23 » жовтня 2019 року

ЗАВДАННЯ

**на магістерську дисертацію
студенту Ярошенку Максиму Олександровичу**

(прізвище, ім'я, по батькові)

1. Тема дисертації Методи підвищення роздільної здатності інфрачервоних зображень за допомогою штучних згорткових нейронних мереж

науковий керівник дисертації: Варфоломєєв А.Ю., к.т.н., доц.,
(Прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «05» листопада 2021 року № 3241-с

2. Термін подання студентом дисертації: 13 травня 2021 року

3. Об'єкт дослідження: процес підвищення роздільної здатності інфрачервоних зображень

4. Предмет дослідження: методи підвищення роздільної здатності інфрачервоних зображень за допомогою штучних згорткових нейронних мереж.

5. Перелік завдань, які потрібно розробити: 1. Аналіз та патентний пошук існуючих методів збільшення роздільної здатності інфрачервоних зображень. 2. Обрання базового методу збільшення роздільної здатності інфрачервоних зображень для детального розгляду та подальшого вдосконалення. 3. Розробка нового або вдосконалення обраного методу збільшення роздільної здатності інфрачервоних зображень. 4. Програмна реалізація розробленого/вдосконаленого методу збільшення роздільної здатності та його порівняння із базовим методом. 5. Розробка стартап-проєкту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

Презентація у форматі PowerPoint.

7. Орієнтовний перелік публікацій: 1 публікація.

8. Консультанти розділів дисертації

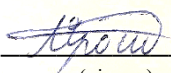
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання: 23 жовтня 2019р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз та патентний пошук існуючих методів збільшення роздільної здатності інфрачервоних зображень	23.10.2019-15.12.2019	Вик.
2	Обрання базового методу збільшення роздільної здатності інфрачервоних зображень для детального розгляду та подальшого вдосконалення	16.12.2019-1.03.2020	Вик.
3	Розробка нового або вдосконалення обраного методу збільшення роздільної здатності інфрачервоних зображень	2.03.2020-1.09.2020	Вик.
4	Програмна реалізація розробленого/вдосконаленого методу збільшення роздільної здатності та його порівняння із базовим методом.	2.09.2020-1.02.2021	Вик.
5	Розробка стартап-проекту	2.02.2021-1.03.2021	Вик.
6	Оформлення дисертації	2.03.2021-13.05.2021	Вик.

Студент


(підпис)

Ярошенко М.О.

(прізвище, ініціали)

Науковий керівник дисертації


(підпис)

Варфоломєєв А.Ю.

(прізвище, ініціали)

РЕФЕРАТ

Магістерська дисертація складається з 92 сторінок, в яких міститься 30 рисунків та 22 таблиці, використано 37 джерел.

Актуальність. Зростаюча розповсюдженість інфрачервоних сенсорів вкупі з їхньою обмеженою роздільною здатністю (більшість рішень на ринку мають не більше 1 мегапікселя) створюють необхідність розробки методів підвищення роздільної здатності. Серед цих методів існують й ті, які базуються на штучних нейронних мережах. Однак, не дивлячись на великі об'єми даних (знімків) накопичених людством та на зростання продуктивності обчислювальної техніки, що призвели до розвитку нейронних мереж, підвищення роздільної здатності для інфрачервоних зображень шляхом застосування нейронних мереж не набуло такого розповсюдження та уваги з боку дослідників, як, наприклад, для зображень у видимому спектрі. Саме тому розробка нових та удосконалення вже наявних методів підвищення роздільної здатності інфрачервоних зображень є важливим та **актуальним** завданням.

Зв'язок роботи з науковими програмами, планами, темами. Матеріали роботи використані в ініціативній НДР (державний реєстраційний номер: 0120U101505) КПП ім. Ігоря Сікорського. Дисертаційні дослідження відповідають науковим напрямкам кафедри КЕОА, а також пріоритетному напрямку розвитку науки і техніки України «Інформаційні та комунікаційні технології».

Метою роботи є підвищення роздільної здатності інфрачервоних зображень шляхом удосконалення існуючих методів на основі згорткових нейромереж та створення алгоритмічних і програмних рішень їх реалізації.

Для досягнення поставленої мети в роботі вирішувалися наступні **задачі**:

- проаналізовано існуючі методи збільшення роздільної здатності для інфрачервоних та зображень з видимого спектру;

- обґрунтовано вибір в якості базового для подальшого удосконалення методу на основі нейромережі BCLSR;
- запропоновано власну нейронну мережу HCNNSR, та шляхом моделювання проведено її порівняльний аналіз з нейромережею BCLSR.

Об’єктом дослідження є процес підвищення роздільної здатності інфрачервоних зображень.

Предметом дослідження є методи підвищення роздільної здатності інфрачервоних зображень за допомогою штучних нейронних мереж.

Методами дослідження є згорткові нейронні мережі.

Наукова новизна отриманих результатів полягає в тому, що:

- Вперше запропоновано використання відомої нейронної мережі BCLSR для збільшення роздільної здатності інфрачервоних зображень, що дозволило шляхом дослідження визначити її ефективність та обґрунтувати її вибір в якості базової для подальшого удосконалення.
- Удосконалено нейронну мережу BCLSR у вигляді ієрархічної згорткової нейромережі для підвищення роздільної здатності інфрачервоних зображень HCNNSR, яка відрізняється від базової реалізації BCLSR відсутністю рекурсивних та рекурентних шарів, що дозволило підвищити майже вдвічі (в 1,9 раз) її швидкодію та забезпечити більш ефективне відновлення зображення (за параметром PSNR різниця сягає 0,063 дБ).

Практичне значення отриманих результатів визначається створеними алгоритмічними та програмними рішеннями реалізації запропонованої нейронної мережі. Створено програмну модель HCNNSR на мові програмування Python засобами бібліотеки Tensorflow в середовищі Google Colaboratory.

Публікації. За матеріалами досліджень підготовлено 2 статті. Опубліковано у фаховому виданні України першу статтю (див. Додаток А):

- Yaroshenko, M. O., Varfolomeiev, A. Y., Yaganov, P. O. Інерційна система розпізнавання жестів. // Мікросистеми, Електроніка та Акустика. 2019. Vol. 24, No. 5. С. 42–47.

Друга ж стаття проходить етап рецензування для публікації у фаховому виданні України:

- Yaroshenko, M. O., Varfolomeiev, A. Y., Yaganov, P. O. Ієрархічна згорткова нейронна мережа для підвищення роздільної здатності інфрачервоних зображень. // Мікросистеми, Електроніка та Акустика. 2021.

Ключові слова: підвищення роздільної здатності зображень; згорткові нейронні мережі; рекурентні нейронні мережі; інфрачервоні зображення.

ABSTRACT

This Master's thesis consists of 92 pages, which contain 30 figures and 22 tables, 37 literature references are used.

Relevance. Growing infrared sensors' market adoption and their limited resolution (the vast majority of present solutions are less than 1 megapixel) create the need for methods for resolution increase (sometimes called super-resolution). There are artificial neural network methods among them. But, despite the high amount of data (images in our case) collected by humans and increasing computational power which caused the neural networks' development, resolution increase of infrared pictured provided by neural networks hasn't gained such popularity and attention from academic community as, for instance, according methods for visible wavelength imagery. That is why development of new methods and improvement of known approaches of resolution increase for infrared images is important and **relevant** task.

Relationship of work with scientific programs, plans, topics. Dissertation's materials are used in the initiative R&D (registration number: 0120U101505) of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute». Thesis's research was conducted in accordance with the scientific activity of the Department of ECED, and with the priority of science and technology of Ukraine "Information and communication technologies".

The aim of the work is to increase the resolution of infrared images by improving existing methods based on convolutional neural networks and creating algorithmic and software solutions for their implementation.

To achieve this goal, during the research the following **tasks** were solved:

- known methods of resolution increase for visible and infrared images were analyzed;
- BCLSR was reasonably chosen as the basic method for further improvement;
- proprietary neural network HCNNSR was proposed, it's comparative analysis with the BCLSR network was conducted by the modelling.

The object of the research is the process of infrared images' resolution increase.

Methods of infrared images' super-resolution based on artificial neural networks are **the subject** of the research.

Methods of the research are convolutional neural networks.

The scientific novelty of the results of the study is that:

- For the first time, the known neural network BCLSR was proposed to increase the resolution of infrared images, which allowed to research its effectiveness and justify its choice as a basis for further improvement.
- The improvement of the BCLSR – a neural network HCNNSR (Hierarchical Convolutional Neural Network for infrared image Super-Resolution) – was proposed, which differs from the BCLSR in absence of the recurrent and recursive layers. The proposed solution is almost twice (1.9 times) faster than the BCLSR and slightly better in infrared images' super-resolution – the difference is 0.0063 dB in terms of the PSNR.

The practical significance of the obtained result is determined by the created algorithmic and software solutions for the implementation of the proposed neural networks. Software model of the proposed HCNNSR network coded in Python programming language with the Tensorflow library and Google Colab environment.

Publications. Two papers based on the materials of the dissertation were prepared. The first one was published in the Ukrainian professional edition:

- M. A. Yaroshenko, A. Yu. Varfolomieiev, P. A. Yaganov. The Inertial Gesture Recognition System. *Microsystems, Electronics and Acoustics*. Kyiv, 2019. № 5. P. 42-47

The second article is going through the reviewing process before publishing in the Ukrainian professional edition:

- M. A. Yaroshenko, A. Yu. Varfolomieiev, P. A. Yaganov. Hierarchical convolutional neural network for infrared image super-resolution. *Microsystems, Electronics and Acoustics*. Kyiv, 2021.

Key words: super-resolution; convolutional neural networks; recurrent networks; far infrared images.

Зміст

Вступ.....	3
РОЗДІЛ 1 АНАЛІЗ НАЯВНИХ МЕТОДІВ ЗБІЛЬШЕННЯ ЗОБРАЖЕНЬ, В ТОМУ ЧИСЛІ Й ІНФРАЧЕРВОНИХ.....	6
1.1 Проблема підвищення роздільної здатності інфрачервоних зображень .	7
1.2 Результати кон'юнктурного дослідження ринку та патентного пошуку	8
1.3 Аналіз існуючих методів на базі нейронних мереж	10
Висновок по розділу 1	22
РОЗДІЛ 2 ШТУЧНА НЕЙРОННА МЕРЕЖА HCNNSR ДЛЯ ЗБІЛЬШЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ІНФРАЧЕРВОНИХ ЗОБРАЖЕНЬ.....	23
2.1 Мережа BiConvLSTM SR (BCLSR).....	23
2.1.1 Блок рекурсивного виводу	24
2.1.2 Двонапрямлені LSTM-шари.....	26
2.1.3 Subpixel (Depth-to-Space).....	28
2.1.4 Глобальний залишковий шлях (Global Residual Path).....	28
2.2 Запропонована мережа – ієрархічна згорткова нейронна мережа для підвищення роздільної здатності (HCNNSR).....	28
2.2.1 Inception-блок	30
2.2.2 Бікубічна інтерполяція	31
Висновки по розділу 2	33
РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ НЕЙРОННОЇ МЕРЕЖІ HCNNSR.....	35
3.1 Особливості навчання.....	35
3.2 Результати тестування мереж	36
Висновки по розділу 3	40
РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ	41
4.1 Опис ідеї проєкту	41

4.2 Техніко-економічна характеристика продукції	42
4.3 Актуальність проєкту та аналіз ринку	44
4.4 Розробка ринкової та маркетингової стратегії продукту	52
Висновок по розділу 4	57
ЗАГАЛЬНІ ВИСНОВКИ	59
Список використаних джерел	61
Додаток А. Копії публікацій у фахових виданнях.....	66
Додаток Б. Лістинг програми.....	80

Вступ

Актуальність. Зростаюча розповсюдженість інфрачервоних сенсорів вкупі з їхньою обмеженою роздільною здатністю (більшість рішень на ринку мають не більше 1 мегапікселя) створюють необхідність розробки методів підвищення роздільної здатності. Серед цих методів існують й ті, які базуються на штучних нейронних мережах. Однак, не дивлячись на великі об'єми даних (знімків) накопичених людством та на зростання продуктивності обчислювальної техніки, що призвели до розвитку нейронних мереж, підвищення роздільної здатності для інфрачервоних зображень шляхом застосування нейронних мереж не набуло такого розповсюдження та уваги з боку дослідників, як, наприклад, для зображень у видимому спектрі. Саме тому розробка нових та удосконалення вже наявних методів підвищення роздільної здатності інфрачервоних зображень є важливим та **актуальним** завданням.

Зв'язок роботи з науковими програмами, планами, темами. Матеріали роботи використані в ініціативній НДР (державний реєстраційний номер: 0120U101505) КПІ ім. Ігоря Сікорського. Дисертаційні дослідження відповідають науковим напрямкам кафедри КЕОА, а також пріоритетному напрямку розвитку науки і техніки України «Інформаційні та комунікаційні технології».

Метою роботи є підвищення роздільної здатності інфрачервоних зображень шляхом удосконалення існуючих методів на основі згорткових нейромереж та створення алгоритмічних і програмних рішень їх реалізації.

Для досягнення поставленої мети в роботі вирішувалися наступні **задачі**:

- проаналізовано існуючі методи збільшення роздільної здатності для інфрачервоних та зображень з видимого спектру;
- обґрунтовано вибір в якості базового для подальшого удосконалення методу на основі нейромережі BCLSR;

- запропоновано власну нейронну мережу HCNNSR, та шляхом моделювання проведено її порівняльний аналіз з нейромережею BCLSR.

Об’єктом дослідження є процес підвищення роздільної здатності інфрачервоних зображень.

Предметом дослідження є методи підвищення роздільної здатності інфрачервоних зображень за допомогою штучних нейронних мереж.

Методами дослідження є згорткові нейронні мережі.

Наукова новизна отриманих результатів полягає в тому, що:

- Вперше запропоновано використання відомої нейронної мережі BCLSR для збільшення роздільної здатності інфрачервоних зображень, що дозволило шляхом дослідження визначити її ефективність та обґрунтувати її вибір в якості базової для подальшого удосконалення.
- Удосконалено нейронну мережу BCLSR у вигляді ієрархічної згорткової нейромережі для підвищення роздільної здатності інфрачервоних зображень HCNNSR, яка відрізняється від базової реалізації BCLSR відсутністю рекурсивних та рекурентних шарів, що дозволило підвищити майже вдвічі (в 1,9 раз) її швидкодію та забезпечити більш ефективне відновлення зображення (за параметром PSNR різниця сягає 0,063 дБ).

Практичне значення отриманих результатів визначається створеними алгоритмічними та програмними рішеннями реалізації запропонованої нейронної мережі. Створено програмну модель HCNNSR на мові програмування Python засобами бібліотеки Tensorflow в середовищі Google Colaboratory.

Публікації. За матеріалами досліджень підготовлено 2 статті. Опубліковано у фаховому виданні України першу статтю (див. Додаток А):

- Yaroshenko, M. O., Varfolomieiev, A. Y., Yaganov, P. O. Інерційна система розпізнавання жестів. // Мікросистеми, Електроніка та Акустика. 2019. Vol. 24, No. 5. С. 42–47.

Друга ж стаття проходить етап рецензування для публікації у фаховому виданні України:

- Yaroshenko, M. O., Varfolomieiev, A. Y., Yaganov, P. O. Ієрархічна згорткова нейронна мережа для підвищення роздільної здатності інфрачервоних зображень. // Мікросистеми, Електроніка та Акустика. 2021.

Структура даної роботи містить вступ, 4 розділи, загальний висновок, перелік використаної літератури та додатки:

- Додаток А. Копії публікацій у фахових виданнях;
- Додаток В. Лістинг програми.

РОЗДІЛ 1 АНАЛІЗ НАЯВНИХ МЕТОДІВ ЗБІЛЬШЕННЯ ЗОБРАЖЕНЬ, В ТОМУ ЧИСЛІ Й ІНФРАЧЕРВОНИХ

В останні десятиліття застосування тепловізорів розширюється завдяки експлуатації наземних, бортових або космічних платформ. Наземні спостереження можуть використовуватися, наприклад, для виявлення витоків енергії в будівлях або моніторингу робочих характеристик в промислових процесах, в той час як аерофотозйомка використовується для підтримки міського планування і точного землеробства і т.д. Всі ці застосування поділяють необхідність точної і докладної інформації про температуру поверхні та її просторову структуру.

Незважаючи на вражаючий технологічний розвиток теплових датчиків, викликаний зростаючим попитом, теплове зображення все ще страждає від обмеженої роздільної здатності, яка гіршою в порівнянні зі звичайними зображеннями, отриманими у видимій смузі або в ближній інфрачервоній області. Виключаючи супутникові датчики, дійсно, в даний час доступні камери, що працюють в довгохвильовому тепловому інфрачервоному діапазоні (8-14 мікрон), які не досягають одного мегапікселя. З апаратної точки зору подальше збільшення кількості пікселів на одиницю площі досить складно, тому що, з одного боку, зменшення розміру пікселя призведе до зниження також відносини сигнал / шум і, з іншого боку, до збільшення розміру датчика. було б занадто дорого для пристроїв загального призначення.

У цьому контексті суперроздільна здатність (SR – super-resolution), що являє собою процес генерування зображення з більш високою роздільною здатністю (HR – high resolution), починаючи з одного або декількох кадрів з більш низьким дозволом (LR – low resolution), представляється багатообіцяючим підходом для подолання поточних технологічних обмежень і забезпечення можливості аналізувати більш дрібні деталі температурних характеристик поверхні [1].

1.1 Проблема підвищення роздільної здатності інфрачервоних зображень

Згідно із [2], головною метою super-resolution є відновлення HR-зображення із LR. Загалом, LR-зображення I_x моделюється як результат деградації \mathcal{D} із деякими параметрами δ (як-от шум або фактор масштабування) відповідного HR-зображення I_y :

$$I_x = \mathcal{D}(I_y; \delta) \quad (1.1)$$

Зазвичай процес деградації (\mathcal{D} та δ) є невідомим, а наявними є лише LR-зображення. В цьому випадку, також відомому як сліпе SR (англ. blind Super Resolution), перед дослідниками постає задача відновити HR-апроксимацію \hat{I}_y оригінального HR-зображення I_y із LR-зображення I_x :

$$\hat{I}_y = \mathcal{F}(I_x; \theta), \quad (1.2)$$

де \mathcal{F} - SR-модель, а θ – параметри \mathcal{F} .

Незважаючи на невизначеність \mathcal{D} та фактори, які на неї впливають (як-от артефакти утискання, шум сенсору або гранульований шум), дослідники намагаються змодельовати деградацію. Більшість робіт моделюють її як одну операцію понижуючої дискретизації (downsampling):

$$\mathcal{D}(I_y; \delta) = (I_y) \downarrow_s, \{s\} \subset \delta, \quad (1.3)$$

де \downarrow_s - операція понижуючої дискретизації із фактором масштабування s . Дійсно, більшість датасетів для типового SR побудовано за цим шаблоном, із використанням бікубічної інтерполяції із анти-аліасінгом як функції понижуючої дискретизації. Втім, в деяких роботах деградація моделюється як комбінація кількох операцій:

$$\mathcal{D}(I_y; \delta) = (I_y * k) \downarrow_s + n_\zeta, \{k, s, \zeta\} \subset \delta, \quad (1.4)$$

де $I_y * k$ позначає згортку HR-зображення із ядром розмиття (англ. blur kernel), n_ζ – деякий адитивний Гаусів шум із стандартним відхиленням ζ . Модель (4) є більш наближеною до реальної ситуації, ніж (3).

Отже, задача SR виглядає таким чином:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\hat{I}_y, I_y) + \lambda \Phi(\theta), \quad (1.5)$$

де $\mathcal{L}(\hat{I}_y, I_y)$ – функція втрат між згенерованим HR-зображенням \hat{I}_y та справжнім зображенням I_y , $\Phi(\theta)$ – доданок регуляризації, λ – компромісний параметр (англ. tradeoff parameter). Зазвичай в якості функції втрат використовується піксельна середньоквадратична похибка, однак, точніші моделі все частіше застосовують декілька функцій втрат.

1.2 Результати кон'юнктурного дослідження ринку та патентного пошуку

Великі виробники тепловізорів почали впроваджувати алгоритми SR в прошивку своїх останніх моделей. Наприклад, FLIR розробила технологію UltraMax, яка заснована на швидкому отриманні 16 теплових кадрів менш ніж за 1 с [3]. Через природні рухи оператора, кожне зображення буде дещо зміщене щодо інших. В результаті виходить нове зображення з подвоєною роздільною здатністю за умови, що користувач або ціль не дуже сильно переміщаються і що в сцені мінімальний контраст [1].

Відомий патент (**WO/2012/072151, Method for preparing images in non-visible spectral ranges, and corresponding camera and measuring arrangement**) на подібне рішення, який базується на суміщенні кількох інфрачервоних кадрів із нечіткою реєстрацією на базі одного опорного кадру. Перевагою цього методу є його безперервність, причому кадри потоку ІЧ-відеоданих, виділені для розрахунку SR-ІЧ-зображення, вибирають з «віконця», що переміщається під час проходження потоку ІЧ-відеоданих. Завдяки такому безперервному виконанню, на пристрої виведення завжди є наготові потік відеоданих SR-ІЧ-зображень, що мають високу чіткість. Недоліками розглянутого методу є необхідність у рухах тепловізійної камери та необхідність робити декілька кадрів низької роздільної здатності для отримання одного SR-ІЧ-зображення.

Замість цього Infratech використовує апаратне рішення, що складається з швидкого обертового колеса MicroScan [4], яке дозволяє отримувати чотири

експозиції, кожне з яких зміщено в поперечному напрямку на половину пікселя.

Інші програмні рішення засновані на об'єднанні даних із зображеннями у видимій смузі. FLIR MSX [5], наприклад, підвищує чіткість зображення шляхом нанесення деталей видимого світла на теплові зображення. Цей процес, однак, імовірно, змінить теплове вимірювання, тому що більш дрібні деталі беруться з видимої смуги і не обов'язково відповідають змінам температури поверхонь, що спостерігаються [1].

Існує патент **CN108961180A (Infrared image enhancement method and system)**, який здійснює збільшення роздільної здатності інфрачервоного зображення шляхом використання інформації про деталі з видимого спектру. Загалом метод складається з трьох кроків:

- 1) Вилучення контурної інформації із видимого зображення високої чіткості, що відповідає інфрачервоному зображенню низької роздільної здатності, використовуючи згорткову нейронну мережу;
- 2) Інфрачервоне зображення низької роздільної здатності обробляється за допомогою нейронної мережі сегментації суперпікселів для отримання інформації інфрачервоного зображення високої чіткості;
- 3) Обробка інформації інфрачервоного зображення високої чіткості та інформації контуру видимого зображення високої чіткості.

Іншим способом отримати кілька інфрачервоних кадрів для SR-задачі є застосування кількох ІЧ-камер. Метод, описаний в патенті **CN104252704A (Total generalized variation-based infrared image multi-sensor super-resolution reconstruction method)** виконує проектування зображень в один координатний простір, їх розрідження, визначення вагових коефіцієнтів та нормалізація отриманого зображення. Як зазначено в патенті, експерименти показують, що якість зображення, реконструйованого запропонованим методом, є високою, а зображення – близьким до оригінального інфрачервоного зображення високої роздільної здатності. Головним

недоліком даного методу є необхідність у використанні кількох ІЧ-сенсорів із низькою роздільною здатністю.

1.3 Аналіз існуючих методів на базі нейронних мереж

Існуючі традиційні алгоритми надвисокої роздільної здатності, такі як інтерполяція, за своєю суттю не генерують відсутню високочастотну інформацію, а просто змішують суміжні пікселі з ваговим співвідношенням, і ефект розмиття і згладжування відновленого зображення більш очевидний. Підхід на основі реконструкції (англ. reconstruction based approach) відновлює зображення шляхом введення ефективних обмежень попереднього знання. Якість відновлення висока, але й складність обчислень висока, а швидкість збіжності низька. Підхід розрідженого вираження (англ. sparse expression) значно кращий порівняно з попередніми двома, але ємність моделі і здатність вираження обмежені, і він поступово був замінений підходом глибокого навчання [6].

У 2015 році була вперше [7] застосована згорткова нейронна мережа SRCNN (Super Resolution Convolutional Neural Network) для реконструкції суперроздільної здатності зображення. SRCNN (Рис. 1.1 [7]) використовує модель CNN для відповідності відображень з низькою і високою роздільною здатністю і використовує велику кількість таких пар зображень навчання моделі.

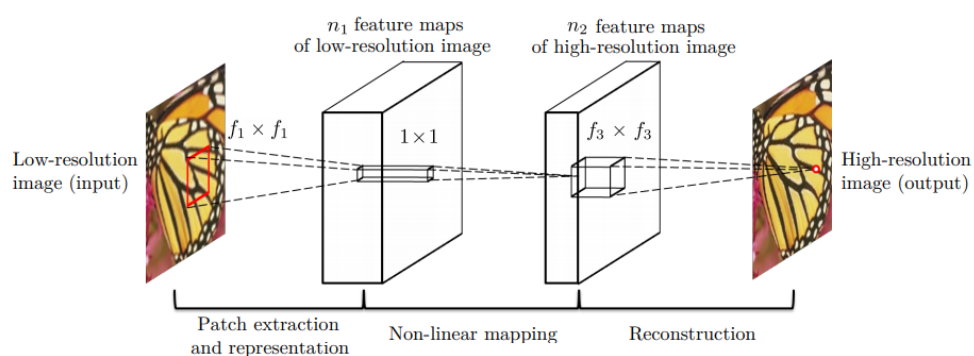


Рисунок 1.1 – Структура SRCNN та призначення згорткових шарів

Результати перевершують відносно просте вкладення околу (neighbor embedding) і розріджене кодування (sparse coding) [7]. Однак тришарова структура SRCNN не може задовольнити вимоги більш високої якості зображення і продуктивності в реальному часі. Відповідно до характеристик згорткової нейронної мережі, перші кілька шарів можуть отримувати тільки поверхневу інформацію про текстуру зображення. Щоб отримати більш детальні характеристики і отримати кращий ефект реконструкції, необхідно використовувати більш глибоку мережу, таку як VDSR (Рис. 1.2) [8], що має 20 згорткових шарів [6].

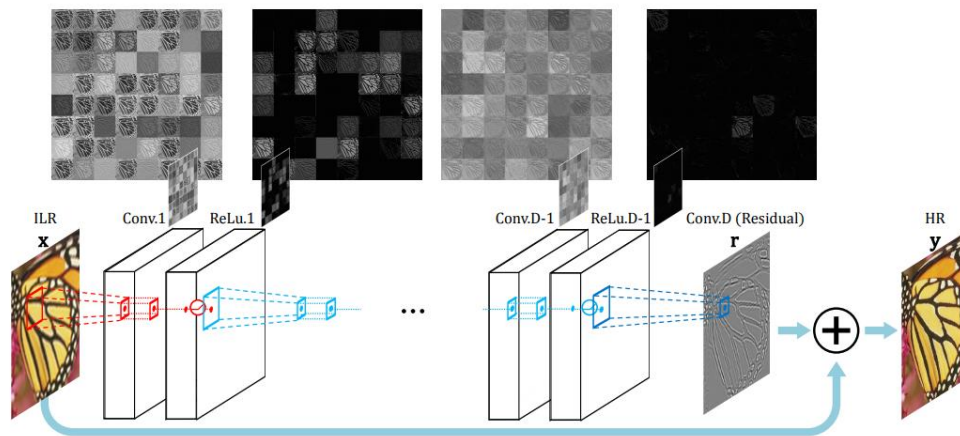


Рисунок 1.2 – Структура VDSR

Зображення з низькою роздільною здатністю в усіх вищезазначених підходах спочатку бікубічно інтерполюється, а потім оброблюється мережею. Для досягнення продуктивності в реальному часі було запропоновано мережу ESPCN (Рис. 1.3) [9], яка безпосередньо витягує зображення з низькою роздільною здатністю і забезпечує її підвищення за рахунок збільшення числа каналів характеристичних ознак (feature channels) [6].

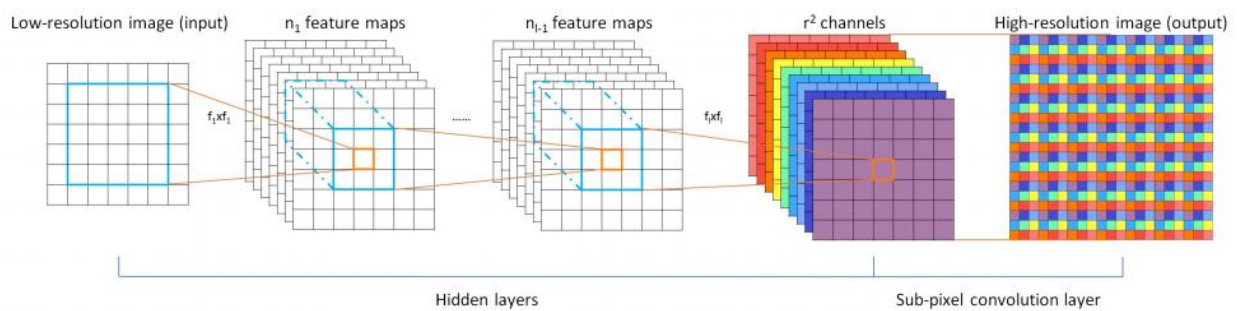


Рисунок 1.3 – Ефективна субпiксельна згорткова нейронна мережа (ESPCN) з двома шарами згортки для вилучення карт об'єктiв i субпiксельним шаром згортки, який агрегує карти об'єктiв з простору LR i створює зображення SR

Подiбно до ESPCN, мережа FSRCNN (Рис. 1.4) [10] для досягнення продуктивностi в реальному часi використовує деконволюцiї або транспонованi згортковi шари. FSRCNN видiзняється від SRCNN в основному в трьох аспектах. По-перше, FSRCNN приймає вихiдне зображення низької роздiльної здатностi в якостi вхiдних даних без бiкубiчної iнтерполяцiї. Рiвень деконволюцiї вводиться в кiнцi мережi для пiдвищення розмiру кадру (upsampling). По-друге, етап нелiнiйного вiдображення в SRCNN замiнюється трьома етапами в FSRCNN, а саме етапом стиснення, вiдображення i розширення. По-третє, FSRCNN використовує фiльтри меншого розмiру i бiльш глибоку структуру мережi. Цi полiпшення забезпечують FSRCNN кращу продуктивнiсть, але меншу обчислювальну вартiсть, нiж SRCNN [6].

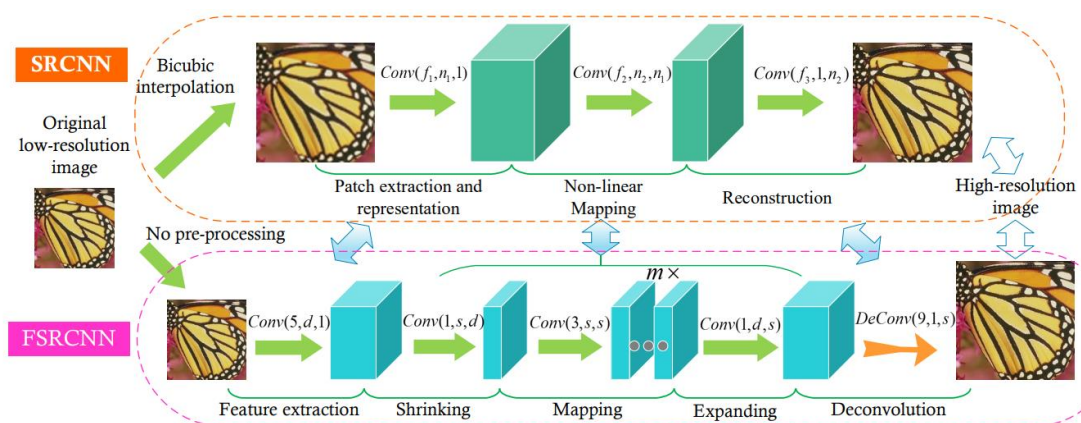


Рисунок 1.4 – Структури SRCNN i FSRCNN

SRGAN (Рис. 1.5) [11] використовує GAN (змагальна породжуюча мережа) для поліпшення деталей зображення. SRGAN, хоч і має значення PSNR (Peak Signal-to-Noise Ratio) нижче за ESPCN, але згенеровані нею зображення мають вищий показник MOS (Mean Opinion Score), тобто на думку незалежних оцінювачів є візуально найбільш якісними та найближчими до оригіналу. У 2017 році LapSRN (Рис. 1.6) [12] об'єднала лапласову піраміду [13] і модель нейронної мережі, використовуючи покрокове навчання з підсиленням для підвищення точності реконструкції суперроздільної здатності [6].

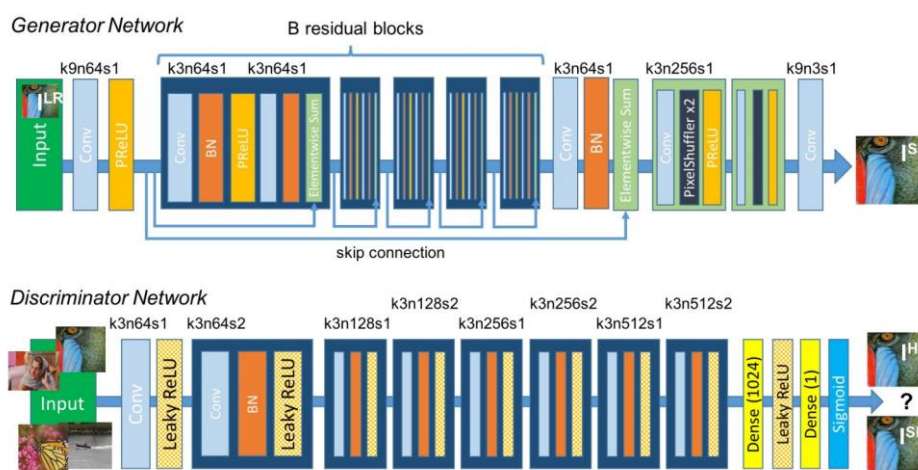


Рисунок 1.5 – Архітектура мережі генератора і дискримінатора з відповідним розміром ядра (k), кількістю карт характеристик (n) і кроком (s), зазначеними для кожного згорткового шару

У 2018 році було запропоновано SRMD (Рис. 1.7) [14], яка вводить ядро розмиття і стандартне відхилення шуму погіршення зображення в якості одного з вхідних шарів, та досягає чіткого і точного несліпого (non-blind) ефекту суперроздільної здатності відновлення краю зображення [6].

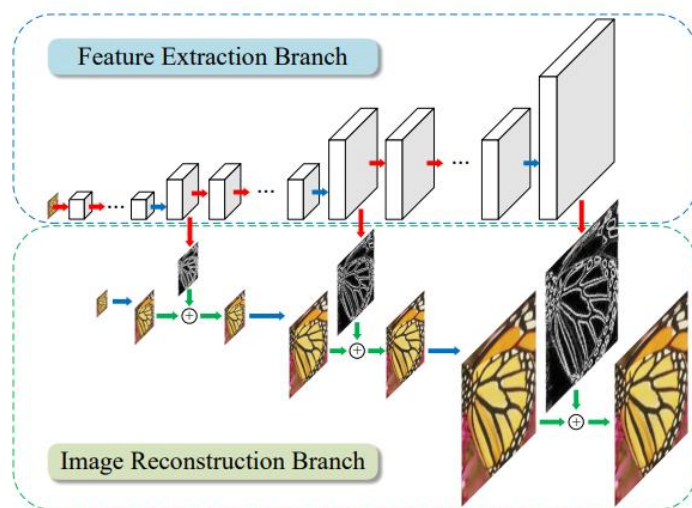


Рисунок 1.6 – Архітектура мережі LapSRN. Червоні стрілки – згорткові шари, сині стрілки - транспоновані згортки, зелені – по-елементні оператори складання

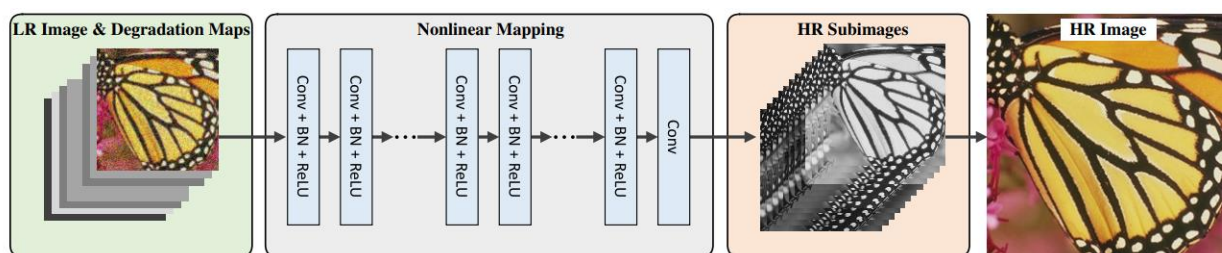


Рисунок 1.7 – Архітектура SRMD

Також у 2018 році представлено SFTGAN (Рис. 1.8) [15], яка застосовує семантичну сегментацію для генерації більш реалістичного зображення більшого розміру за допомогою GAN [6].

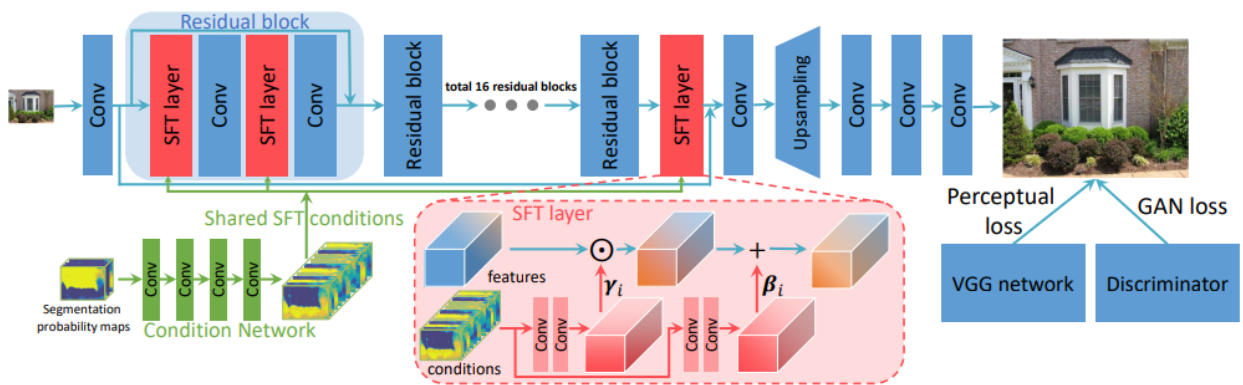


Рисунок 1.8 – Архітектура мережі SFTGAN. Роботу SFT-шару (Spatial Feature Transform) спрямовано на корекцію зображення виходячи з наявності відповідних семантичних ознак (наприклад, виявлення текстури трави на зображенні з метою заміни її на більш якісну)

Підхід на основі CNN для поліпшення теплових зображень був введений в (Рис. 1.9) [16], натхненний пропозицією в [7]. Автори [16] порівнюють точність мережі, навченої різних спектрах зображень, щоб мати найкраще уявлення про покращення теплових зображень. Вони прийшли до висновку, що мережа, навчена на відтинках сірого, забезпечила кращий результат, ніж навчена на теплових зображеннях MWIR-спектру (Mid-wavelength infrared) [17].

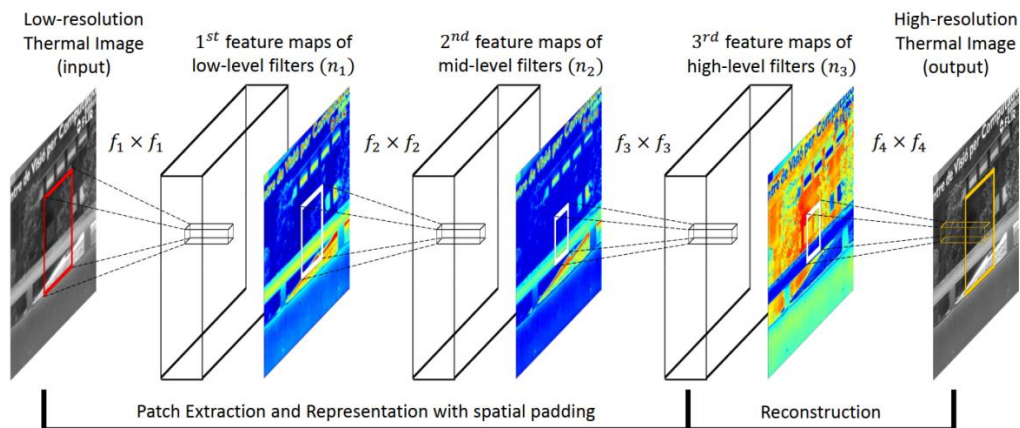


Рисунок 1.9 – Архітектура мережі TEN (Thermal Image Enhancement using Convolutional Neural Network)

З іншого боку, в [18] запропоновано згорткову нейронну (Рис. 1.10) мережу для поліпшення теплових зображень, для навчання якої методом залишкового навчання (residual learning), автори оцінюють ефективність чотирьох кольорових представлень, а саме: відтінки сірого, яскравість (L в просторі HSL), інтенсивність (I в просторі HSI) і V (з кольорового простору HSV). Автори приходять до висновку, що V-представлення є найкращим для поліпшення теплових зображень, а запропонований ними підхід також покращує швидкість збіжності [17].

В [17], на відміну від [16] та [18], запропоновано нейромережу (Рис. 1.11, [17]), навчену виключно на теплових зображеннях, яка показала кращі результати, ніж дві попередні архітектури.

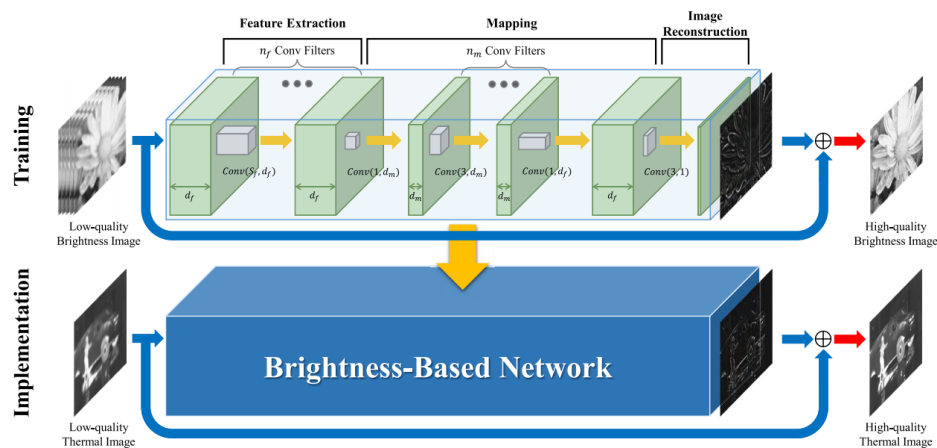


Рисунок 1.10 – Архітектура нейромережі на базі аналізу яскравості

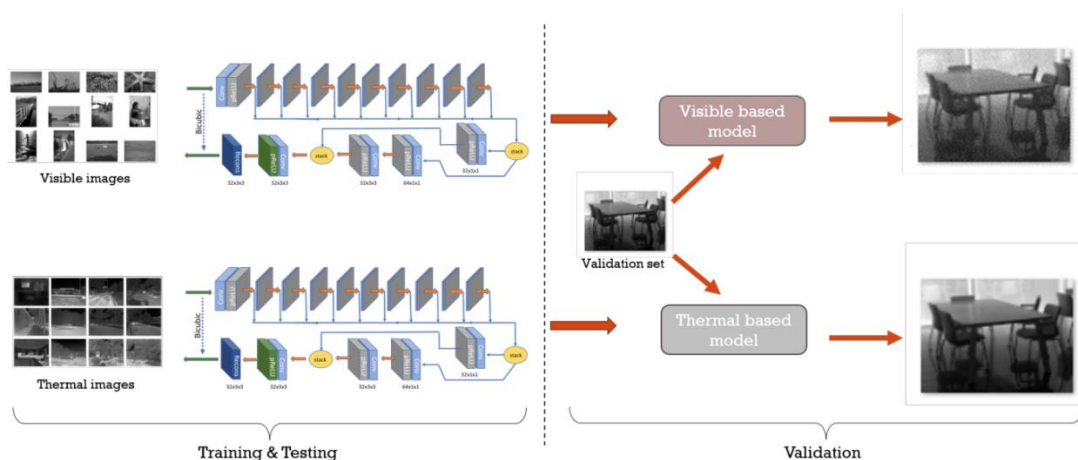


Рисунок 1.11 – Архітектура мережі та порівняння двох моделей із різною навчаючою вибіркою

Розглянуті до цього моменту архітектури нейронних мереж склалися в основному зі згорткових шарів. Втім, глибока модель, тобто створена шляхом додавання більшої кількості згорткових шарів, CNN (Convolutional Neural Network) призводить до появи величезної кількості параметрів та ускладнює процедуру навчання. Збільшення глибини нейронних мереж за рахунок додавання більшої кількості згорткових шарів підвищує і ризик перенавчання [19]. Для боротьби із цим існує 3 основні підходи (Рис.12, [19]):

- Пропуск шарів (Рис.1.12, а): як глобальне залишкове з'єднання (residual connection), яке пов'язує вхідні дані і вихідний шар, як, наприклад, в [8, 15, 17, 18], так і локальне залишкове з'єднання з ResNet [20], зменшують труднощі в навчанні.
- Структури з щільними зв'язками (dense connection structures) пов'язують всі шари в мережі і об'єднують карти об'єктів всіх попередніх рівнів (Рис. 1.12, b), щоб зменшити зникаючі градієнти і повторно використовувати елементи з дрібних шарів (shallow layers).
- Рекурсивні нейронні (Рис. 1.12, c) мережі неодноразово застосовують одні й ті ж згорткові шари для керування параметрами з досягненням великого поля сприйняття [19].

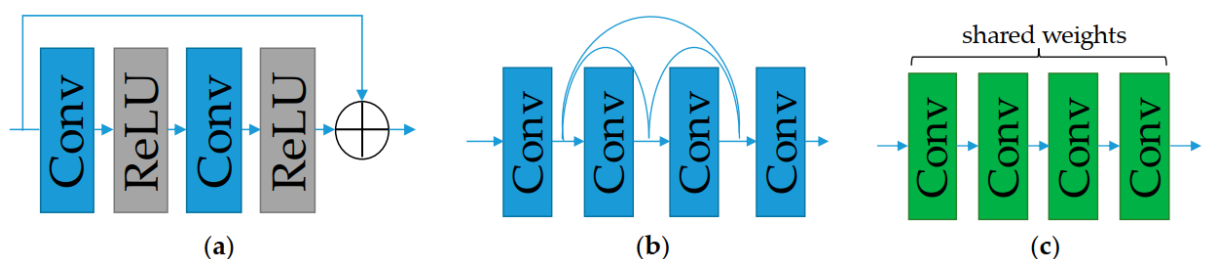


Рисунок 1.12 – Основні підходи до зменшення глибини нейронної мережі: а) пропуск шарів; б) структури з щільними зв'язками; c) рекурсивні нейронні мережі (зеленим позначено рекурсивні згорткові шари)

У задачі SR рекурентні нейронні мережі зазвичай використовуються для захоплення довгострокової залежності сусідніх кадрів в відео [21]. BRCN (Рис. 1.13) [22, 23] складається з трьох частин: згортковий шар з прямим зв'язком для захоплення просторової залежності між LR і HR, двонапрямлена рекурентна згорткова мережа для захоплення тимчасової залежності між послідовними кадрами і умовний згортковий шар для подальшого захоплення просторово-часових залежностей [19].

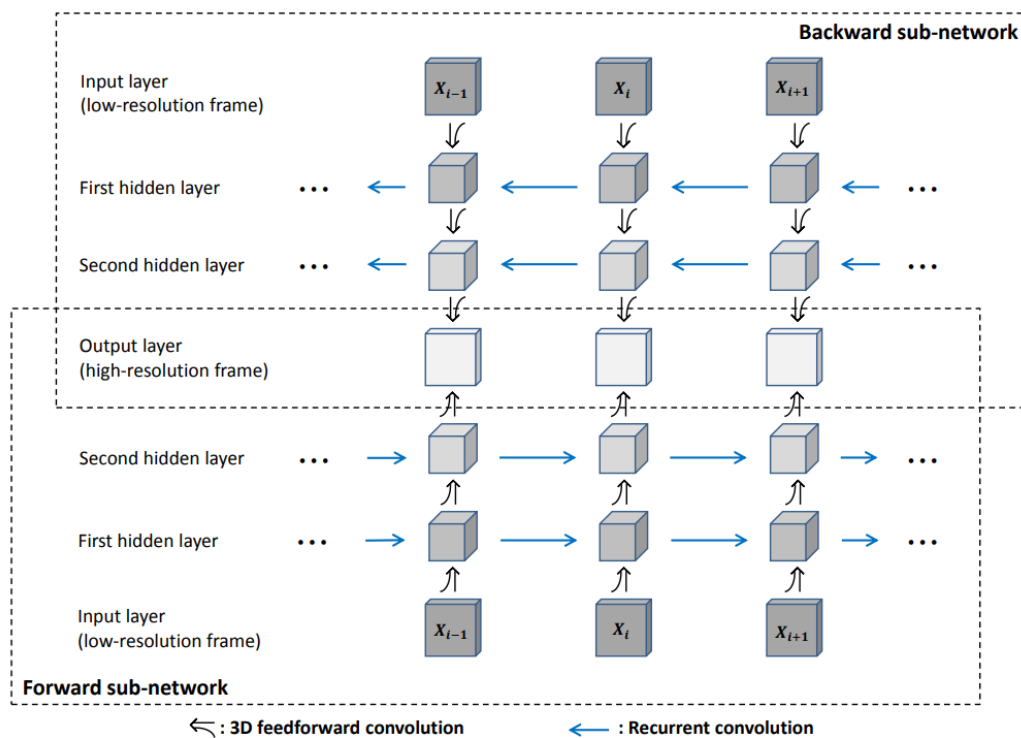


Рисунок 1.13 – Архітектура BRCN

STCN [24] пропонує двосторонню структуру LSTM для захоплення просторово-часової інформації для відновлення відеокadra. Для вирішення задачі SR одиночних зображень, заснованого на розгляді ResNet в якості розгорнутої RNN, DSRN (Рис. 1.14) [25] пропонує рекурентну мережу з двома станами, і кожен стан працює з просторовою роздільною здатністю LR і HR окремо, щоб досліджувати з'єднання між парами LR і HR і забезпечує потік інформації від LR до HR при кожній рекурсії.

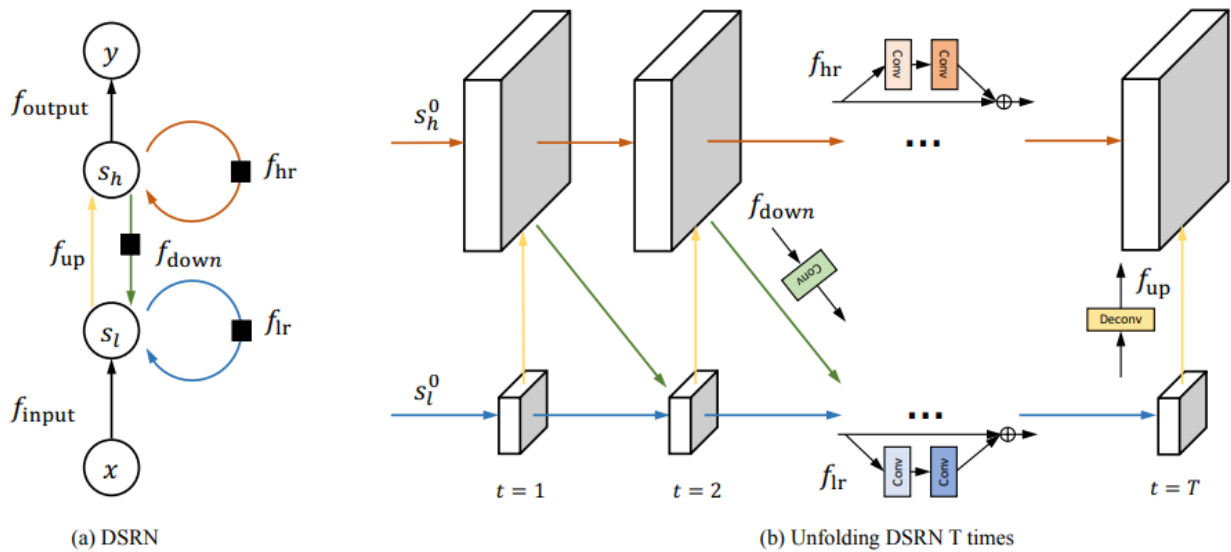


Рисунок 1.14 – а) Архітектура DSRN; б) Розгорнута структура DSRN. Ребра з тим же кольором, що в а), мають ідентичні функції перехідного стану і параметри. Структури чотирьох конкретних функцій переходу були проілюстровані відповідно. Блоки Conv з різними кольорами вказують на різні параметри.

В [19] запропоновано нейронну мережу (Рис. 1.15), що складається з двох структур:

- 1) Блок рекурсивного виводу (Recursive inference block (RIB) на Рис. 1.15), заснований на повторному використанні щільних з'єднань локальних низькорівневих характеристик, із застосуванням рекурсивного навчання для управління параметрами моделі при одночасному збільшенні рецептивних полів;
- 2) Двонаправлений згортковий шар LSTM (BiConvLSTM на Рис. 1.15), для визначення кореляції ознак з кожної рекурсії і адаптивного вибору додаткової інформації для шару реконструкції.

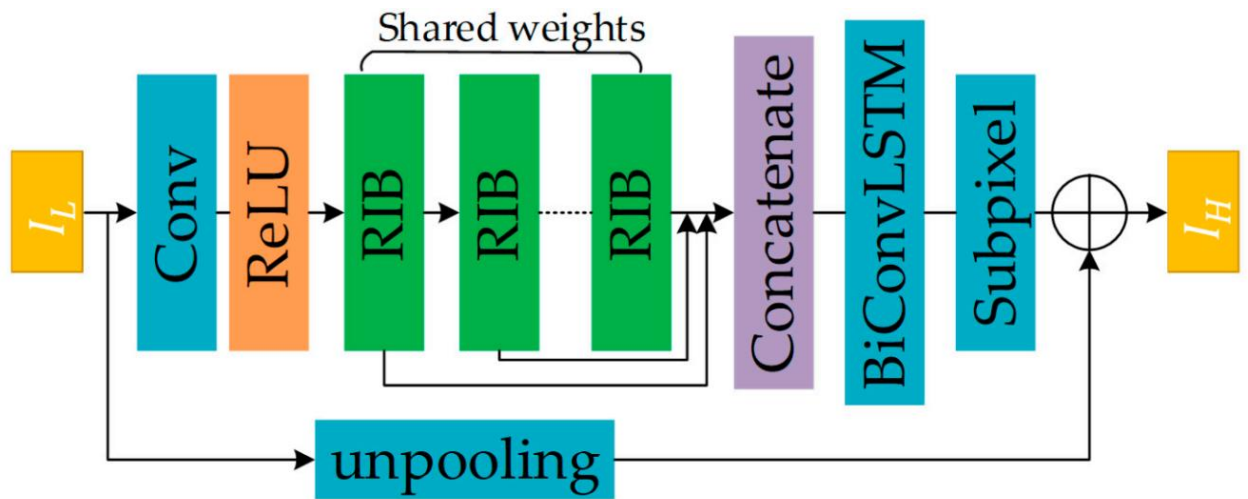


Рисунок 1.15 – Архітектура BCLSR

Візуальне порівняння розробленої в [19] мережі наведено на Рис.1.16.

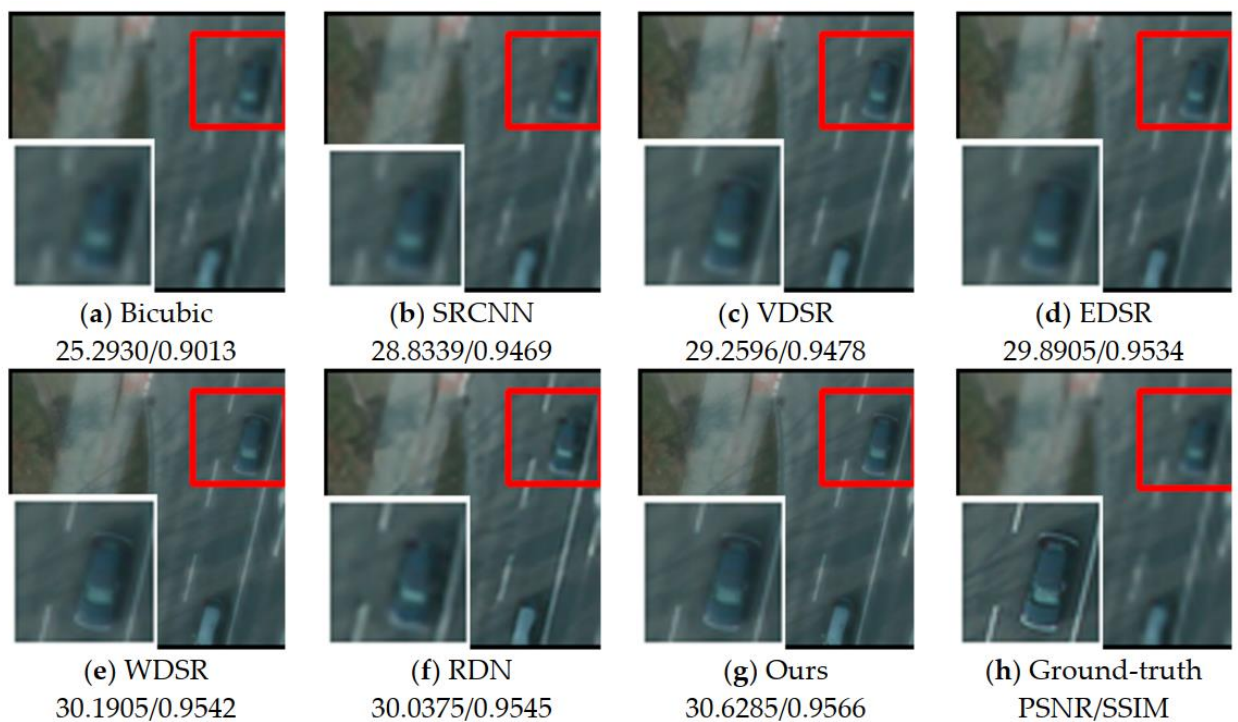


Рисунок 1.16 – Візуальне порівняння роботи популярних неймереж для вирішення SR-задачі (Ours – мережа BCLSR)

Автори BCLSR стверджують, що запропонована ними мережа перевершує аналоги за рахунок таких аспектів:

- 1) Через збільшення поля сприйняття з кожною рекурсією, більше контекстної інформації може використовуватись для відновлення якісного зображення без додаткових параметрів.
- 2) Завдяки повторному використанню низькорівневих ознак (англ. low-level features), потік інформації через мережу може бути підсилений без «вибуху» або «зникнення» градієнту.
- 3) Двонаправлений згортковий LSTM-шар вибірково витягує доповнюючу інформацію з усіх рекурсій та запобігає надходженню зайвої інформації до шару відновлення.

В [19] заявлено, що BCLSR є кращою за такі відомі SR-мережі, як WDSR, RDN, EDSR та ін. Графічне порівняння параметрів наведено на Рис. 1.17 (випробувано на датасеті COWC [26]), Рис. 1.18, [19]. Таким чином, нейромережі на базі LSTM-шарів на даний момент є кращими за якістю відновлення високої роздільної здатності, складністю навчання та кількістю параметрів. Однак не виявлено жодної згадки про застосування LSTM-шарів до обробки інфрачервоних зображень. Саме тому BCLSR покладено в основу розробки цільової мережі даної роботи.

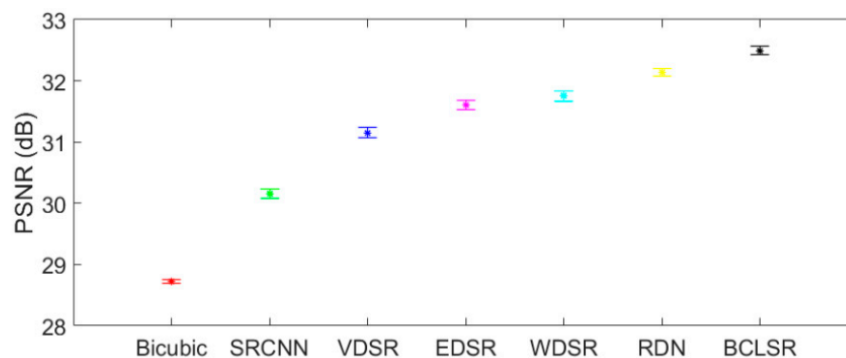


Рисунок 1.17 – Середнє значення та середньоквадратичне відхилення PSNR із фактором масштабування 4

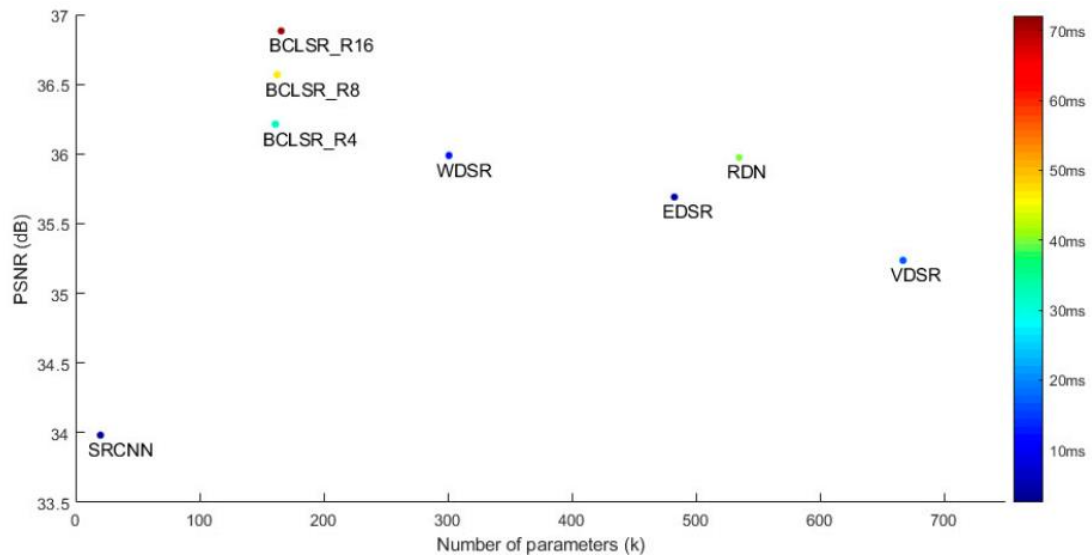


Рисунок 1.18 – Порівняння PSNR та розмірів моделей для SR на датасеті COWC із коефіцієнтом масштабування 2. Кольори крапок, що відповідають кольорам на шкалі справа, вказують на час обробки зображення 256x256 пікселів на наявних в авторів BCLSR графічних прискорювачах

Висновок по розділу 1

В даному розділі наведено результати патентного пошуку та аналіз відомих методів підвищення роздільної здатності інфрачервоних зображень та зображень з видимої смуги. Через вдалі архітектурні рішення, та, як наслідок, високу якість збільшення роздільної здатності при малих розмірах, обрано нейронну мережу BCLSR як основу для дослідження та розробки власного методу реконструкції високої роздільної здатності інфрачервоних зображень.

РОЗДІЛ 2 ШТУЧНА НЕЙРОННА МЕРЕЖА HCNNSR ДЛЯ ЗБІЛЬШЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ІНФРАЧЕРВОНИХ ЗОБРАЖЕНЬ

Через описані в минулому розділі переваги BCLSR, цю нейронну мережу обрано за основу запропонованого методу підвищення роздільної здатності інфрачервоних зображень з метою перевірки її архітектурних особливостей на FIR-зображеннях та можливості удосконалення або адаптації для цього класу зображень. Отже, варто розглянути основні елементи її архітектури більш детально. Наступним кроком є розробка власної архітектури із врахуванням виявлених недоліків BCLSR.

2.1 Мережа BiConvLSTM SR (BCLSR)

Згідно із Рис. 1.15 в минулому розділі, BCLSR складається із таких модулів: згортковий шар для роботи із низькорівневими ознаками зображення, блок рекурсивного виводу (RIB – Recursive Inference Block) для виявлення ієрархічних ознак, двонапрявлений згортковий LSTM-шар для злиття ієрархічних ознак та субпіксельний шар для перетворення мап ознак в кінцеве зображення.

Через високу кореляцію вхідного LR- та кінцевого HR-зображень, позначених I_L та I_H відповідно, додано глобальний остаточний шлях F_G (unpooling на Рис. 1.15) із інтерполяцією по найближчому сусіду (nearest neighbor interpolation).

Згортковий шар для роботи із низькорівневими ознаками має розмір ядра 3x3 пікселі та ReLU як функцію активації:

$$F_0 = \max(0, I_L * W_s + b_s), \quad (2.1)$$

де «*» - тут і надалі – згортка, W_s – набір ваг згорткового шару, b_s – його зсув, F_0 – результат роботи цього згорткового шару.

F_0 використовується як вхід до блоку рекурсивного виводу:

$$F_n = S_{RIB}^n(F_0), \quad (2.2)$$

де RIB^n – застосування n -вкладеного блоку рекурсивного виводу, а F_n ($n = 1, \dots, N$) – вихід RIB.

Двонаправлений згортковий LSTM-шар отримує на вхід конкатенацію усіх результатів роботи RIB:

$$F_{fuse} = BCL([F_n, F_{n-1}, \dots, F_1]), \quad (2.3)$$

де BCL – застосування двонаправленого згорткового LSTM-шару.

Потім конкатенація усіх станів останнього двонаправленого згорткового LSTM-шару пропускається через субпіксельний шар (Subpixel на Рис.1.15) та додається до інтерпольованого F_G зображення.

2.1.1 Блок рекурсивного виводу

Як зазначалося раніше, одним з основних елементів мережі є блок рекурсивного виводу (Рис. 2.1 [19]) – розроблений під впливом архітектури іншої мережі – ResNet [20], що застосовується для класифікації зображень. В цілому, використання компонентів із мереж для класифікації у мережах, що призначені для підвищення роздільної здатності, є виправданим, оскільки припускається, що в обох випадках використовуються одні й ті самі ознаки вхідних зображень. Однак, результат їх виділення є різним: відгуки згорткових шарів під час класифікації збуджують нейрони, що відповідають різним класам, тоді як в задачах підвищення роздільної здатності вони є основою для додавання деталей до зображення.

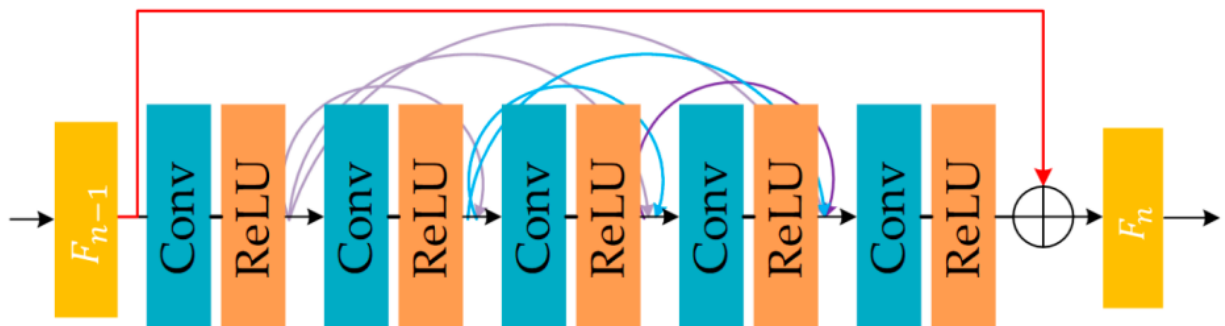


Рисунок 2.1 – Блок рекурсивного виводу мережі BCLSR. F_{n-1} та F_n – вихід попередньої ітерації та вхід наступної відповідно

Згідно із [19], перша згортка застосовується для розширення ознак (feature expansion). Ядро розміром 1×1 використовується для збільшення інформації, що проходить через шар, що широко застосовується в мережі ResNet для збільшення або зменшення каналів. Для збереження нелінійності застосовано ReLU як функцію активації:

$$F_{n,0} = \max(0, F_{n-1} * W_e + b_e), \quad (2.4)$$

де W_e – набір ваг згорткового шару, b_e – його зсув, F_{n-1} – вихід минулої рекурсії. $F_{n,0}$ потім використовується як вхід частини RIB із щільними зв'язками (англ. dense connections).

Декілька згорткових шарів, входи яких є конкатенаціями усіх попередніх згорткових шарів, відокремлюються авторами BCLSR в окремий шар щільного з'єднання (densely connected layer):

$$F_{n,i} = \max(0, [F_{n,0}, F_{n,1}, \dots, F_{n,i-1}] * W_{n,i} + b_{n,i}), \quad (2.5)$$

де W_e – набір ваг згорткового шару, b_e – його зсув, $[F_{n,0}, F_{n,1}, \dots, F_{n,i-1}]$ – конкатенація попередніх згорткових шарів.

Таке з'єднання зміцнює потік інформації, що проходить через RIB, повторно використовує ознаки, вилучені в попередніх шарах, запобігаючи повторному вивченню ознак, та, таким чином, зменшує проблему зникаючого градієнту [27].

Останній згортковий шар в RIB розробниками BCLSR називається шаром злиття ознак (feature fusion layer), та використовується для підтримання однакової кількості каналів у кожному виході RIB (F_n). Конкатенація усіх входів в останній згортковий шар, а, отже, його вихід, матиме кількість каналів $D \times G + E$, де D – кількість згорткових шарів у шарі щільного з'єднання, G – кількість каналів в кожному з них, E – кількість каналів у шарі розширення ознак. Для збільшення ємності (capacity) RIB та покращення потоку інформації, додано локальний залишковий (residual) шлях між входом та виходом RIB, тобто, вихід n -вкладеного RIB матиме вигляд:

$$F_n = F_{n-1} + \max(0, [F_{n,0}, F_{n,1}, \dots, F_{n,d}] * W_f + b_f), \quad (2.6)$$

де d – кількість згорткових шарів в RIB до шару злиття ознак, W_f – ваги шару злиття ознак, b_f – його зсув.

Авторами BCLSR обрано 1x1 як розмір ядра шару злиття ознак з метою вибіркового злиття ознак з минулих етапів обробки в поточній ітерації RIB.

Отже, робота кількох робота RIB формулюється як:

$$F_n = S_{RIB}^n(F_0) = S_{RIB}(S_{RIB}(\dots S_{RIB}(F_0))). \quad (2.7)$$

2.1.2 Двонапрямлені LSTM-шари

Окремо варто розглянути роботу двонапрямлених LSTM-шарів. Оригінальні LSTM-шари запропоновано в 1997 в [28] з метою збереження інформації про довготривалі залежності між послідовними даними. Цей тип шарів застосовується для обробки та прогнозування часових рядів. Також автором цієї роботи було досліджено класифікацію рухів нейронною мережею на базі LSTM-шарів [29]. В BCLSR використано згорткові LSTM-комірки [30], що мають наступні рівняння виходів та станів:

$$\begin{aligned} i_t &= \sigma(x_t * W_{xi} + h_{t-1} * W_{hi} + b_i) \\ f_t &= \sigma(x_t * W_{xf} + h_{t-1} * W_{hf} + b_f) \\ o_t &= \sigma(x_t * W_{xo} + h_{t-1} * W_{ho} + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(x_t * W_{xc} + h_{t-1} * W_{hc} + b_c) \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (2.8)$$

де,

- « \odot » - тут і надалі – по-елементне перемноження;
- $\sigma(\cdot)$ – сигмоїдна фнкція
- X_t – поточний (t-й) елемент оброблюваної послідовності;
- W_{mn} – ваги для згортки з m при розрахунку вентилію n ($m = \{x, h\}$, $n = \{i, f, o, c\}$);
- b_n – зсув у вентилі n ($n = \{i, f, o, c\}$);

- вентиль «забування» f_t – визначає яку частину інформації з попередньої ітерації відкинути при обробці поточного елементу послідовності;
- вхідний вентиль i_t ;
- вихідний вентиль o_t ;
- стан комірки c_t – є відповідальним за збереження довготривалих залежностей при обробці послідовностей;
- вихід комірки h_t .

У BCLSR використовується два BiConvLSTM шари, кожен з яких містить по дві згорткові LSTM комірки, через які послідовності сигналів пропускаються в прямому та зворотному напрямках (Рис. 2.2), а вихідну послідовність першого BiConvLSTM шару додатково оброблено 3x3 згортковим шаром. Вихід другого BiConvLSTM шару зконкатеновано по виміру каналів та подано на Subpixel шар.

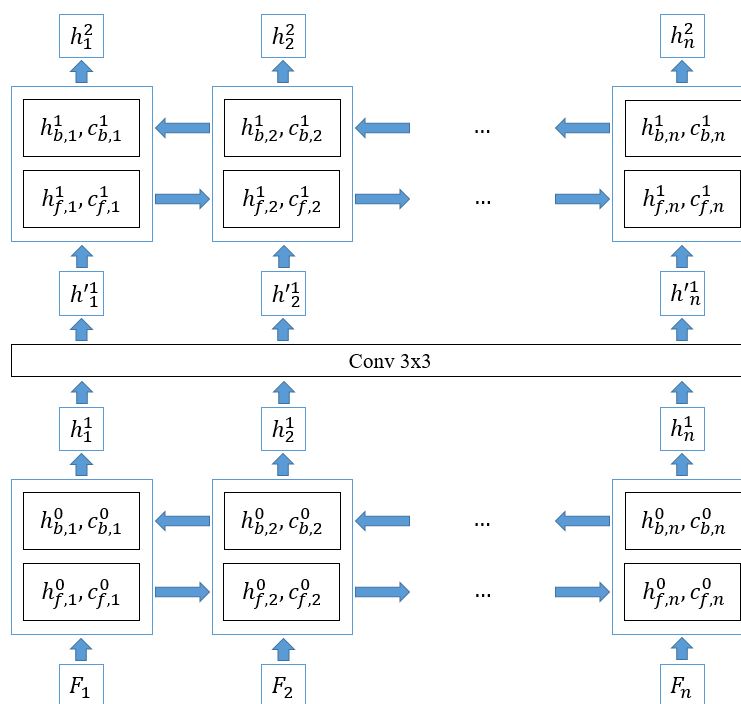


Рисунок 2.2 – Структура двох BiConvLSTM шарів у складі BCLSR. Рекурсії $[F_1, F_2, \dots, F_n]$ сконкатеновано у часову послідовність. h_*^1 та h_*^2 позначають виходи першого та другого шарів відповідно, виходи ConvLSTM-комірок прямого та зворотного напрямку позначено $h_{f,*}^i$ та $h_{b,*}^i$ відповідно

2.1.3 Subpixel (Depth-to-Space)

Для узгодження за розмірами, перед додаванням до інтерпольованого вхідного зображення, вихід попереднього шару спочатку оброблюється згорткою, щоб отримати вектор розрядності $[h_{LR}, w_{LR}, c \times f_{upscale}^2]$, а потім об'єднується в один вектор розрядності $[h_{LR} \times f_{upscale}, w_{LR} \times f_{upscale}, 1]$ як показано на Рисунок 2.3 [31].

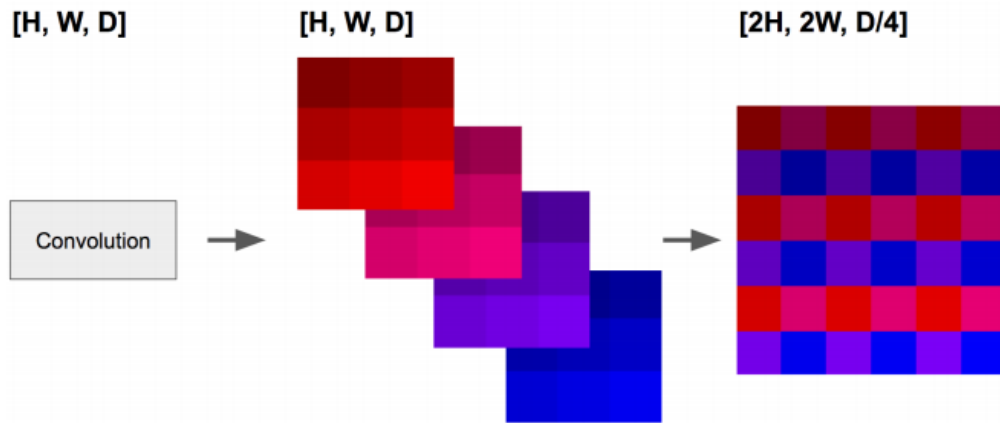


Рисунок 2.3 – Приклад операції Depth-to-Space

2.1.4 Глобальний залишковий шлях (Global Residual Path)

В умовах високої кореляції LR-зображення та HR-зображення, додавання глобального залишкового шляху запобігає вивченню мережею складного повного перетворення вхідного зображення у вихідне. Отже, мережа має лише навчитися додавати високочастотні деталі до вхідного зображення. За твердженнями розробників BCLSR, метод збільшення зображення (upsampling) в глобальному залишковому шляху має зберігати основну структуру та низькочастотну інформацію та мати невелику складність розрахунку. Саму тому було обрано метод найближчого сусіда як upsampling-метод.

2.2 Запропонована мережа – ієрархічна згорткова нейронна мережа для підвищення роздільної здатності (HCNNSR)

BCLSR за даними, наведеними в оригінальній статті [19], дійсно перевершує попередні відомі нейронні мережі для підвищення роздільної

здатності, однак вона має деякі недоліки: по-перше, застосування рекурсивного блоку хоч і зменшує кількість параметрів для навчання та дозволяє регулювати глибину нейронної мережі, втім, таке рішення може мати меншу ємність (capacity of neural network – тобто, в даному випадку, різноманіття та складність образів, які нейронна мережа може коректно обробити), ніж застосування нейронної мережі із такою самою глибиною, але різними параметрами шарів, по-друге, схожий недолік має й рекурентний LSTM шар – не дивлячись на те, що при розрахунку кожної ітерації поточне вхідне значення та значення минулої ітерації «комбінуються» в різних пропорціях, протягом обробки вхідної послідовності значення ваг не змінюються зі збільшенням глибини, що також впливає на гнучкість нейронної мережі.

Для подолання вказаних недоліків було розроблено власну мережу HCNN SR (Hierarchical Convolutional Neural Network for Super Resolution), архітектура якої наведена на Рис. 2.4.

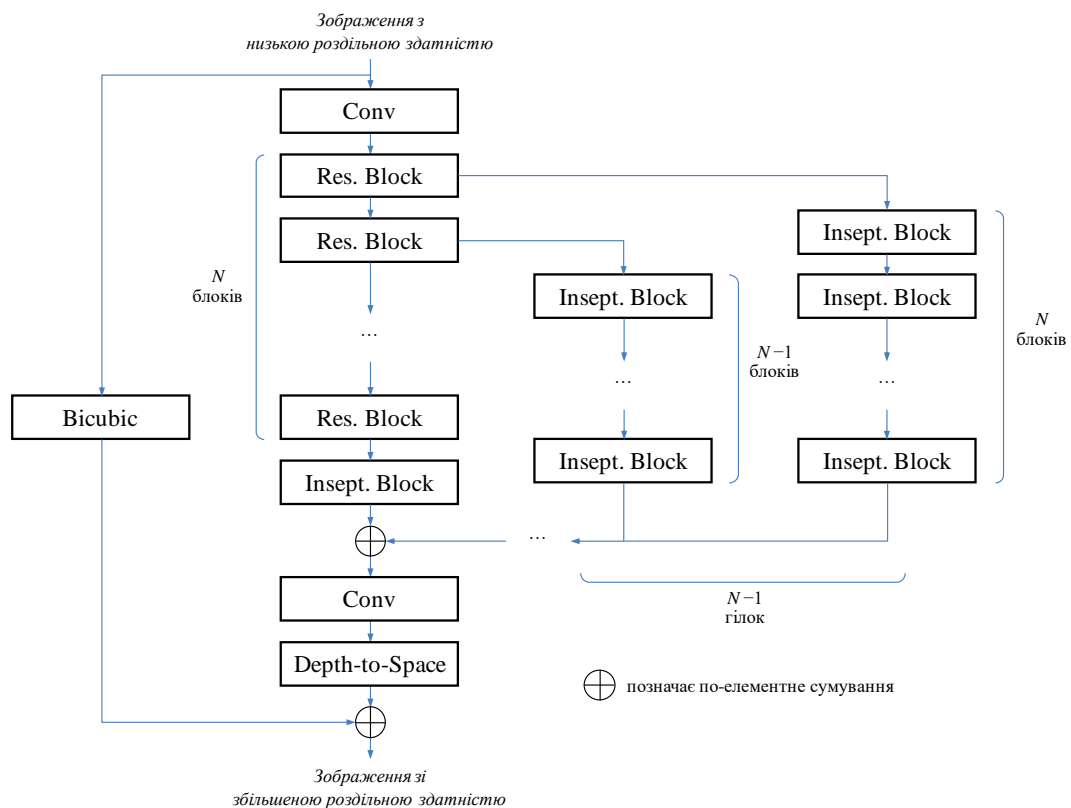


Рисунок 2.4 – Архітектура запропонованої мережі

Архітектура має деревоподібну (ієрархічну) структуру, а вихід кожного із послідовних Residual блоків (структура даного блоку аналогічна RIB блоку, наведеному на Рис.2.1, на Рис. 2.4 позначений як Res.block) утворює гілку, що складається із послідовно з'єднаних Inception-блоків (на Рис. 2.4 позначений як Incept.block), структура яких розглядається нижче. Виходи усіх гілок сумуються і до отриманої суми застосовується перетворення Depth-to-Space. Наступним і останнім кроком є додавання результату останньої операції до вхідного зображення, розширеного за допомогою бікубічної інтерполяції. Таку архітектуру обрано із припущенням, що кожен наступний Residual-блок працює із ознаками різного рівня складності.

2.2.1 Inception-блок

Для подальшої обробки ознак, вилучених після кожного Residual-блоку, як вже зазначалося раніше, створюється окрема «гілка». Кожна така «гілка» складається із послідовно з'єднаних однакових за структурою блоків (Рис. 2.5), створених на основі однієї з найкращих нейронних мереж для розпізнавання зображень Inception [32], та, відповідно, позначених як Incept.block.

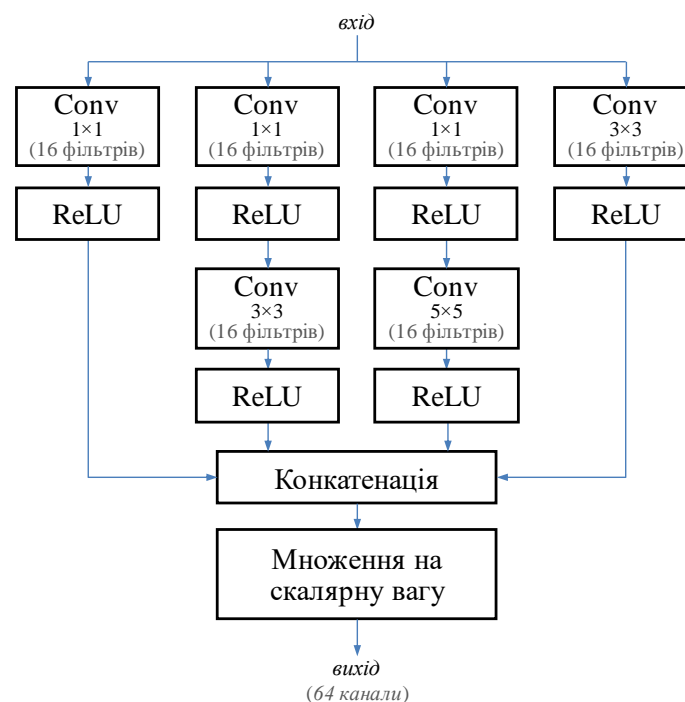


Рисунок 2.5 – Запропонований Inception-блок

Окремо варто розглянути призначення згорток 1×1 , оскільки може бути неочевидним їхнє застосування через мале перцептивне поле. Одиничний розмір ядра дозволяє фільтру працювати як один нейрон із входом з усіх пікселів в одній позиції в усіх мапах ознак (feature map). Тобто, така згортка є просто сумою усіх вхідних каналів із деякою вагою.

В кінці кожного Inception-блоку додано множення на скалярну вагу. Оскільки мережа є ієрархічною, то припускається, що кожен Residual-блок оброблює образи та деталі різних рівнів. Результат кожної такої обробки потім проходить через гілку Inception-блоків. Із кожним подальшим блоком у гілці вилючаються ознаки все вищого рівня, однак, ймовірно, ознаки деяких рівнів потребують підсилення/послаблення, що і виконує шар множення на скалярну вагу.

Після кожного із згорткових шарів в Inception блоках додано dropout шари. За умови невеликої кількості прикладів у навчальних вибірках з FIR зображеннями, додавання dropout шару має запобігти перенавчанню мережі.

2.2.2 Бікубічна інтерполяція

В глобальному залишковому шляху використовується бікубічна інтерполяція – розширення методу кубічної інтерполяції для двовимірних регулярних дискретних просторів. Менше спотворює кінцеве зображення порівняно із іншими популярними методами інтерполяції (Рис. 2.6, [33]).

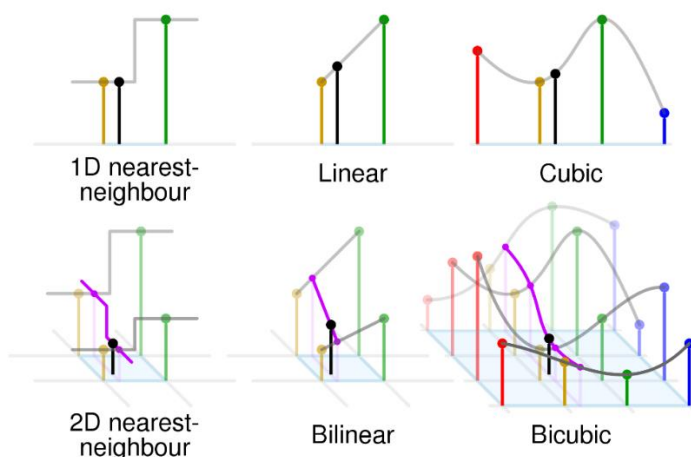


Рисунок 2.6 – Візуальне порівняння методів інтерполяції

В загальному випадку інтерпольоване безперервне зображення можна описати функцією згортки:

$$\hat{v}(x, y) = \sum_i \sum_j v(i, j) g(x - ih, y - jh), \quad (2.9)$$

де $g(t, s)$ – інтерполююча функція (ще називається інтерполяційним ядром), h - шаг дискретизації вихідного зображення, $v(ih, jh)$ – відомі точки зображення.

Найбільш підходящим для інтерполяції зображень є кубічний В-сплайн. Однак відомо, що він має тенденцію до згладжування порівняно із оригінальним SR-зображенням. Саме тому було дослідниками докладено зусиль для обрання кубічного сплайна, що є кращим за кубічний В-сплайн. В загальному випадку кубічний сплайн описується як:

$$g(s) = \begin{cases} a_{30}|s|^3 + a_{20}|s|^2 + a_{10}|s| + a_{00}, & |s| \leq 1 \\ a_{31}|s|^3 + a_{21}|s|^2 + a_{11}|s| + a_{01}, & 1 < |s| \leq 2, \\ 0, & 2 < |s| \end{cases} \quad (2.10)$$

Втім, існує кілька обмежень на це інтерполяційне ядро. Так, необхідно, щоб значення інтерполюючої функції в нулі дорівнювало 1, а в точках з координатами 1 та 2 – дорівнювало 0. Крім того, ядро має бути безперервним в точках 1 та 0, нахил в точках 0 та 2 дорівнював 0, а перша похідна була неперервною. Загалом, це дає 7 обмежень, в той час як невідомих параметрів – 8. Отже, потрібна ще одна умова для однозначного визначення інтерполяційного ядра. Зокрема, якщо інтерполяційну функцію привести у відповідність до перших трьох членів її розкладання в ряд Тейлора, то невідомий параметр α має дорівнювати $\left(-\frac{1}{2}\right)$. Для зручності, усі сім коефіцієнтів визначити через невідомий параметр α , а інтерполяційне ядро визначити як:

$$g(s) = \begin{cases} (\alpha + 2)|s|^3 - (\alpha + 3)|s|^2 + 1, & |s| \leq 1 \\ \alpha|s|^3 - 5\alpha|s|^2 + 8\alpha|s| - 4\alpha, & 1 < |s| \leq 2, \\ 0, & 2 < |s| \end{cases} \quad (2.11)$$

Коли константа α від'ємна, то ядро (2.11) додатне в інтервалі від 0 до 1 та від'ємне на інтервалі від 1 до 2. Таким чином, з від'ємним значенням вільної константи α інтерполяційне ядро має вигляд усіченої sinc-функції (Рис. 2.7 [34]), яка вважається найкращим інтерполянтом у випадку збільшення роздільної здатності, оскільки інтерполяція по суті є оберненою до дискретизації, а, отже, для передискретизації (тобто відновлення) зображення, до нього потрібно застосувати НЧ-фільтр, який «вириже» основну ділянку з обмеженим спектром з-поміж її копій, що повторюються.

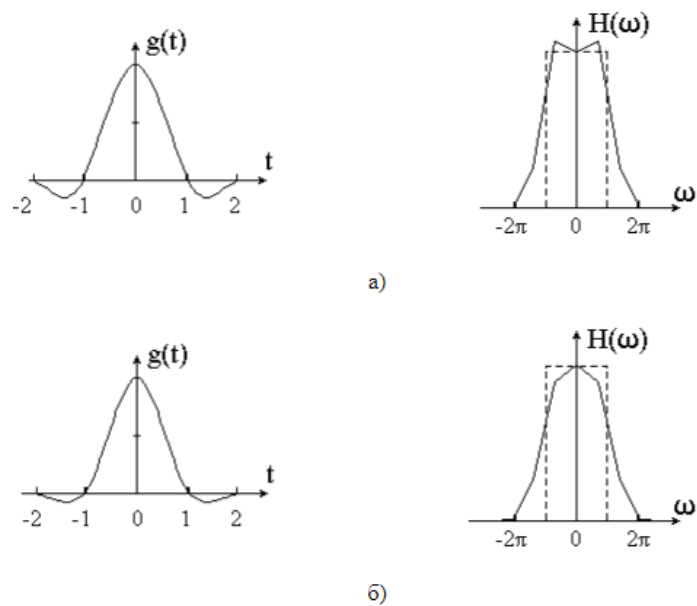


Рисунок 2.7 – Інтерполяційний кубічний сплайн та його частотна характеристика: а) $\alpha = -1$; б) $\alpha = -0,5$;

Для двовимірного випадку інтерполянт має вигляд:

$$g(s, t) = g(s)g(t). \quad (2.12)$$

Висновки по розділу 2

В даному розділі детально розглянуто загальну структуру мережі BCLSR, виявлено недолік, а саме: потенційно обмежена гнучкість через використання рекурсивних та рекурентних шарів. З метою зберегти вдалі архітектурні рішення BCLSR та уникнути вищезгадані недоліки, було

запропоновано мережу HCNNSR, яка не містить рекурсивних та рекурентних шарів, забезпечує окрему обробку ознак зображення різних рівнів, а також використовує більш складний, але й більш точний алгоритм бікубічної інтерполяції на відміну від інтерполяції по найближчому сусіду в BCLSR.

РОЗДІЛ 3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ НЕЙРОННОЇ МЕРЕЖІ HCNNSR

Моделювання запропонованої нейронної мережі виконано засобами програмного пакету Tensorflow в середовищі Google Colaboratory (Рис. 3.1). Досліджувався варіант запропонованої мережі із 8 гілками ($N = 8$ на Рис. 2.4) яка збільшує роздільну здатність в 2 рази. В усіх подальших порівняннях застосовується BCLSR із 8 застосуваннями рекурсивного блоку, яку також реалізовано та навчено засобами Tensorflow для цього дослідження. Для наближення умов порівняння із запропонованим рішенням, згорткові LSTM-шари також використовують dropout. В RIB та Residual блоках встановлено однакову кількість каналів в згорткових шарах – 64.

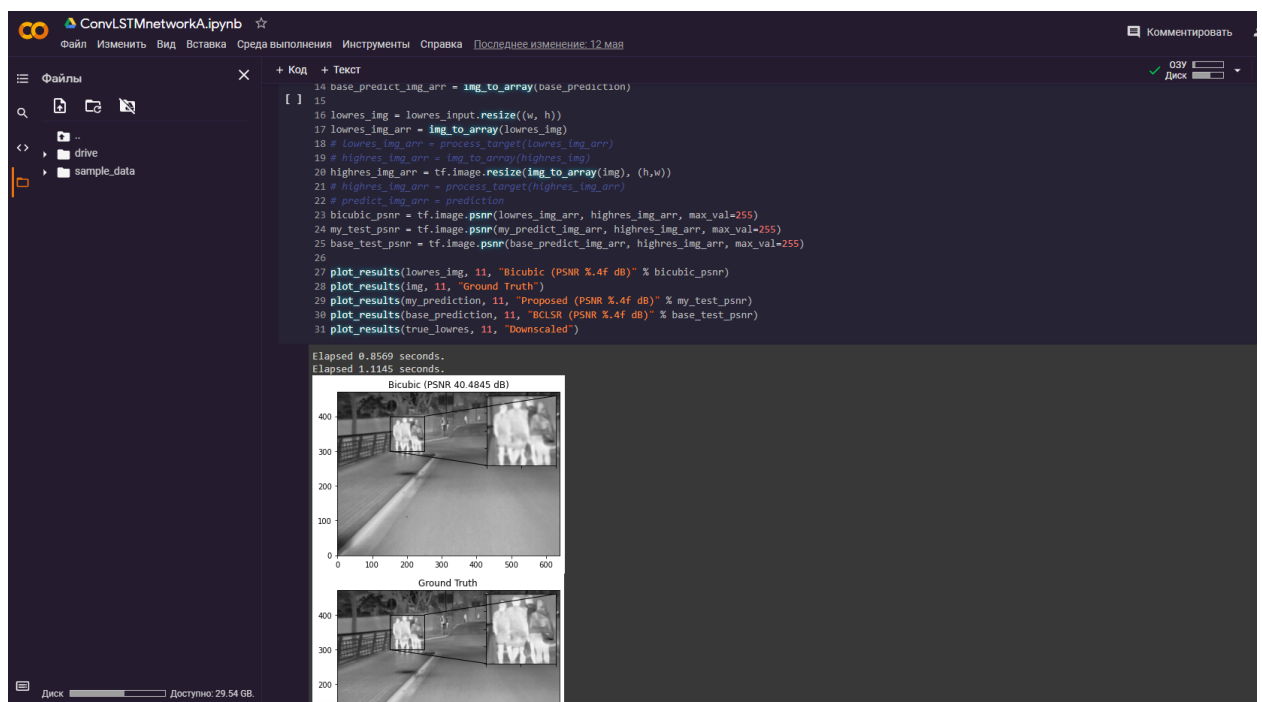


Рисунок 3.1 – Фрагмент інтерфейсу середовища Google Colaboratory під час підготовки зображень до візуального порівняння якості реконструкції

3.1 Особливості навчання

Для навчання та тестування запропонованої мережі та BCLSR використано FIR-зображення, зняті вночі, з навчальної вибірки CVC-14 [36].

Дана вибірка не є великою, тому, задля її розширення, з кожного зображення випадковим чином було вирізано 20 шматків розміром 100×100 , до значень яскравості яких було додано випадкові числа та застосовано випадкову зміну контрастності (функції `random_brightness` та `random_contrast` пакету Tensorflow), до деяких зображень також застосовано віддзеркалення. Отримана навчальна вибірка поділяється на підвибірки (mini-batch) розміром по 8 зображень.

Для оптимізації моделі застосовано оптимізатор Adam із $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$. Функція втрат $L = \frac{1}{mn} \|f - f_{GT}\|^2$ – середньоквадратична похибка між зображенням, що генерує мережа f та еталонним зображенням f_{GT} з високою роздільною здатністю $m \times n$ пікселів.

У роботі [37] показано доцільність циклічної зміни швидкості навчання (одного з параметрів оптимізатора), подібна політика застосована і під час навчання запропонованої мережі – швидкість навчання зростає з 10^{-5} до 10^{-4} за 10 епох, потім так само спадає, додатково після цього одну епоху вона дорівнює 9.1×10^{-6} . Графік зміни швидкості навчання наведений на Рис. 3.2.

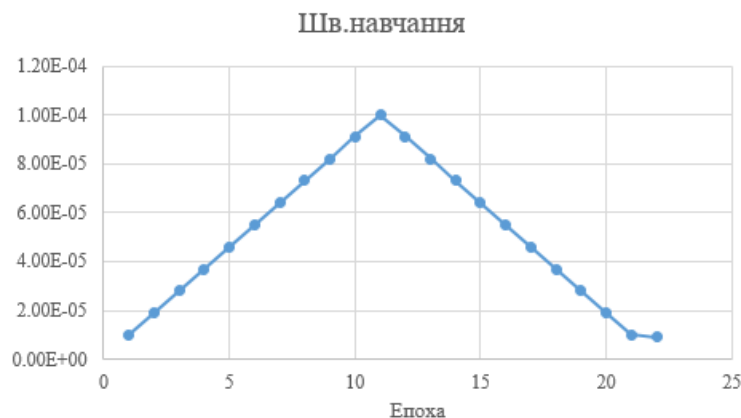


Рисунок 3.2 – Графік зміни швидкості навчання

3.2 Результати тестування мереж

Після 1 циклу навчання отримано нейронну мережу, що відновлює зображення із високою роздільною здатністю із середнім PSNR на тестовому

наборі (FIR-зображення 640×480 пікселів) в 42,2749 дБ, в той час як для бікубічної інтерполяції цей показник дорівнює 40,5721 дБ. Таким чином, маємо приріст PSNR в 1,7028 дБ.

Навчена аналогічним чином мережа BCLSR на тому ж тестовому наборі нічних FIR-знімків CVC-14 забезпечує середній PSNR в 42,2119 дБ. Тобто дає приріст PSNR в 1,6398 дБ порівняно з бікубічною інтерполяцією. Таким чином, якість зображень відновлених запропонованою мережею є дещо вищою в порівнянні з BCLSR. Візуальні результати відновлення для одного з зображень тестової вибірки CVC-14 показано на Рис. 3.3.

Запропонована мережа є помітно більшою за BCLSR. Так, зокрема, вона має 2697512 параметрів, тоді як BCLSR – 591940 параметрів. Не зважаючи на це, запропоноване рішення є більш швидкодіючим: час обробки одного зображення на відеокарті NVIDIA Tesla P100-PCI-E-16GB для нього становить 0,126 с, хоча для BCLSR це значення дорівнює 0,24 с. Причиною цього є відносна складність операцій в LSTM шарах мережі BCLSR – як вже зазначалося, згортковий LSTM шар включає кілька згорток, по-елементне множення, підсумовування та, що важливо, обчислення сигмоїдних функцій та гіперболічних тангенсів (2.8). Загальне порівняння основних характеристик наведено на Рис. 3.4-3.6.



Рисунок 3.3 – Візуальне порівняння оригінального зображення, його утиснутого варіанту (для покращення сприйняття збільшеного методом найближчого сусіда), відновленого BCLSR, відновленого запропонованою нейронною мережею («Proposed») та відновленого бікубічною інтерполяцією

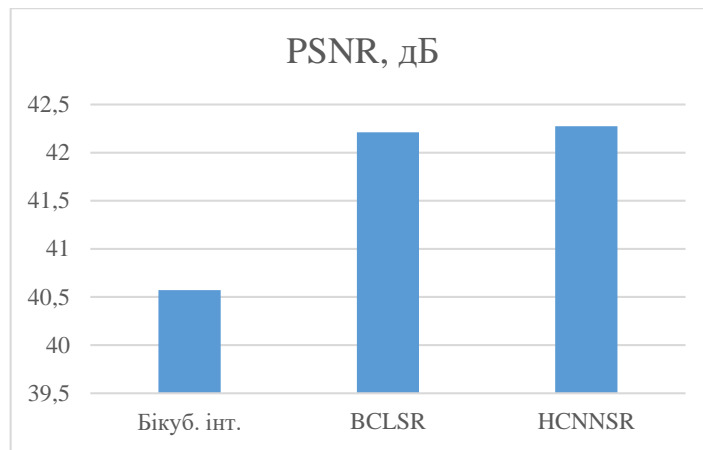


Рисунок 3.4 – Порівняння середньої якості відновлення зображень на тестовому датасеті CVC-14 бікубічною інтерполяцією, мережами BCLSR та HCNNSR

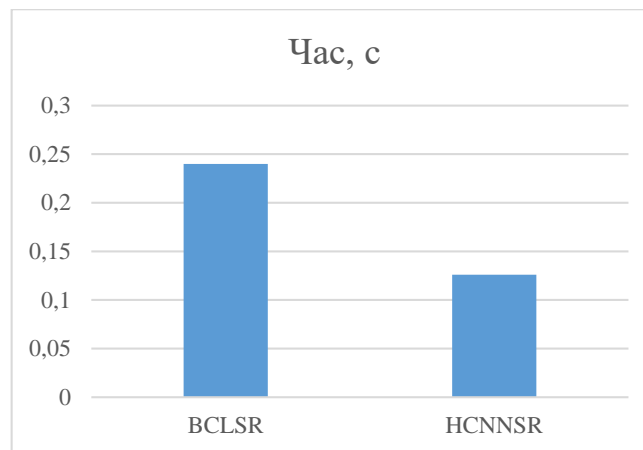


Рисунок 3.5 – Порівняння часу обробки на відеокарті NVIDIA Tesla P100-PCIЕ-16GB одного зображення розміром 640×480 пікселів мережами BCLSR та HCNNSR

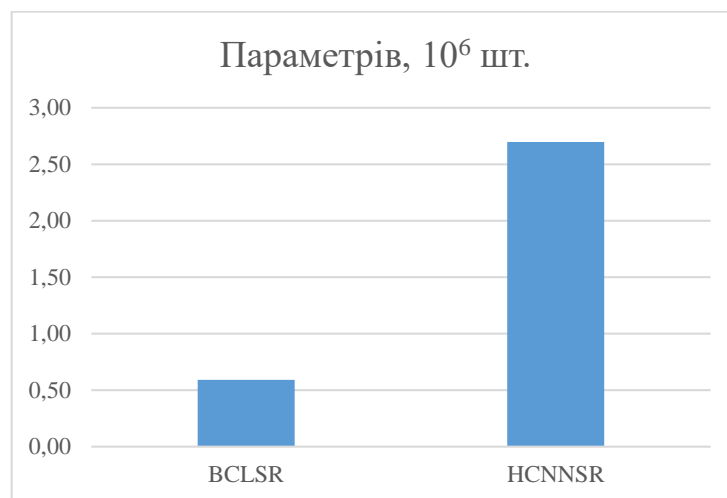


Рисунок 3.6 – Порівняння кількості параметрів мереж BCLSR та HCNNSR

Висновки по розділу 3

В даному розділі наведено умови та результати навчання базової BCLSR з 8 застосуваннями рекурсивного блоку та HCNNSR із 8 «гілками» обробки ознак зображення. При порівнянні навчених мереж виявлено, що, хоч HCNNSR є набагато більшою за розміром, однак вона випереджає базову мережу BCLSR майже в 2 рази за швидкістю, а також трохи (на 0,063 дБ для датасету CVC-14) точніше відновлює роздільну здатність зображень.

РОЗДІЛ 4 РОЗРОБКА СТАРТАП-ПРОЄКТУ

4.1 Опис ідеї проєкту

Відносно висока ціна на інфрачервоні камери створює необхідність у розробці методів підвищення роздільної здатності інфрачервоних зображень. Сучасні найбільші виробники ІЧ-обладнання пропонують різні способи розв'язання цієї проблеми – «злиття» інфрачервоного та зображення у видимій смузі, застосування складних оптичних пристроїв або створення одного зображення з кількох, що отримано через фізіологічно обумовлене тремтіння рук оператора та ін. Втім, такі методи, зокрема вищезгадані, мають суттєві недоліки – невідповідність деталей видимого та ІЧ-зображень, висока ціна та складнощі із застосуванням на нерухомих платформах.

Запропоноване рішення розв'язує проблему збільшення роздільної здатності, готове до використання на різних платформах – мобільних, вбудованих, стаціонарних та ін. – а також має високу швидкодію та є технічно невибагливим. В таблиці 4.1 наведені потенційні напрямки використання проєкту.

Таблиця 4.1 – Напрямки використання стартап-проєкту

Апаратні засоби	Сфери застосування	Напрямки застосування
1. Мобільне обладнання (супутники, наземна техніка, переносні камери та ін.) 2. Обладнання для обробки даних – персональні комп'ютери, центри обробки даних тощо.	1. Системи безпеки. 2. Контроль виробничих процесів. 3. Кліматичні спостереження. 4. Тепловий контроль інфраструктури. та ін.	1. Обробка зображення перед розпізнаванням, відстеженням об'єктів. 2. Візуальне покращення зображення для людського ока.

Низка технологічних параметрів, наведених в табл. 4.2, забезпечують якість та швидкість збільшення роздільної здатності ІЧ-зображень.

Таблиця 4.2 – Технологічні рішення

Ідея	Реалізація	Наявність технології
Висока якість кінцевого ІЧ-зображення із високою роздільною здатністю	Побудова власної нейронної мережі із урахуванням кращих відомих архітектурних рішень	Наявна
Висока швидкість відновлення високої роздільної здатності	Відмова від рекурентних та рекурсивних шарів в мережі	Наявна
Можливість встановлення на платформи різних типів	Реалізація збільшення роздільної здатності ІЧ-зображень виключно програмними засобами	Наявна

Висновки: головна ідея стартап-проєкту формується як: «Швидка мультиплатформенна система збільшення роздільної здатності ІЧ-зображень».

4.2 Техніко-економічна характеристика продукції

В табл. 4.3 наведено фактори стартап-проєкту, які можуть забезпечити йому перевагу порівняно з конкурентами. До технічних факторів віднесено точність збільшення роздільної здатності, час створення одного SR-зображення, мультиплатформеність та потреба в кількох кадрах для створення одного SR-зображення, до економічних: собівартість, можливість

підтримки та обслуговування. Показники *W*, *N*, *S* вказуються на значущість параметра: *W* (weak) – слабка, *N* (neutral) – нейтральна, *S* (strong) – сильна.

Таблиця 4.3 –Визначення слабких, нейтральних та сильних характеристик

№, n/n	Характ-и ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Проект	Конкур-т 1	Конкур-т 2	Конкур-т 3			
1.	Точність	42.27 дБ	42.21 дБ	42.35 дБ	41.7 дБ			+
2.	Час створення SR- зображення	0.126 с	0.24 с	0.58 с	0.56 с			+
3.	Мульти- платфор- меність	є	є	нема	нема			+
4.	Потреба в кількох кадрах	нема	нема	є	є			+
5.	Собівартість	5	5	20	40		+	
6.	Можливість оновлення	є	є	є	нема	+		
7.	Обслуго- вування	є	є	є	є	+		

Висновки: Конкурентні пропозиції мають перевагу в точності відновлених зображень, однак вони повільніші, потребують більше одного кадру на формування SR-зображення та не є мультиплатформеними. Через те, що технічно стартап-проект створено на базі нейронної мережі, це значно зменшує ціну для споживача.

4.3 Актуальність проєкту та аналіз ринку

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проєкту

<i>№ п/п</i>	<i>Показник</i>	<i>Стан показника</i>
1.	Кількість найбільших компаній	3
2.	Загальний обсяг продаж, грн/ум.од	8000
3.	Динаміка ринку	Зростає
4.	Специфічні вимоги до стандартизації	Немає
5.	Середня норма рентабельності, %	70
6.	Наявність обмежень для входу	Немає

Висновок: за показниками ринку можна стверджувати, що ринок зростає, норма рентабельності є доволі високою, не дивлячись на відсутність обмежень для входу, кількість конкурентів є невисокою, втім, вони вже давно на ринку, тож мають досвід, ресурси та потужні технологічні можливості.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проєкту

<i>№ n/n</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія або сегмент ринку</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1.	Потреба в ІЧ-знімках високої якості	Державний та приватний сектор	Інтеграція з різними платформами, різні об'єкти спостереження, необхідність обробки зображень на різних пристроях (обробка в камері або на ПК)	Точність відновлення SR- зображень, швидкодія, легкість у використанні, невисока ціна, підтримка

Висновок: аналіз потенційних клієнтів стартап-проєкту показав, що існує потреба в державному та приватному секторах в підвищенні роздільної здатності ІЧ-зображень на різних платформах із такими параметрами: точність, швидкодія, легкість використання, невисока ціна.

Таблиця 4.6 – Фактори загрози

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст</i>	<i>Можлива реакція кампанії</i>
1.	Якісний	Неналежна якість або підтримка розробленого продукту	Виявлення помилок, розробка кращих підходів
2.	Економічний/ політичний	Економічна та/або політична нестабільність країни-виробника	Перенесення виробництва в іншу країну
3.	Конкурентний	Потужний вплив на ринок відомих гравців	Проведення маркетингової кампанії, підвищення якості продукту

Висновок: Серед розглянутих загрозливих факторів найбільш потужними є конкуренти, які закріпилися на ринку, та якість розроблюваного продукту, що, однак, є доволі типовим для будь-яких нових компаній.

Таблиця 4.7 – Фактори можливостей

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст</i>	<i>Можлива реакція кампанії</i>
1.	Мультиплатформеність	Адаптація технології для роботи на різних платформах	Співпраця із розробниками платформ

Таблиця 4.7 – Фактори можливостей (продовження)

<i>№ n/n</i>	<i>Фактор</i>	<i>Зміст</i>	<i>Можлива реакція кампанії</i>
2.	Покращення точності	Додавання нових датасетів для кращого навчання нейронної мережі	Покупка та/або створення нових датасетів
3.	Індивідуальні замовлення	Підлаштування мережі під специфічні замовлення	Побудова контактів із великими корпоративними клієнтами
4.	Поглинання інших стартапів	Включення нових ідей та кадрів в свою компанію, або прибирання потенційних конкурентів	Моніторинг нових гравців на ринку

Висновок: Розглянуті фактори можливостей є доволі потужними та чисельними для збільшення ресурсів компанії та покращення її основного продукту.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

<i>Особливості конкурентного середовища</i>	<i>В чому проявляється дана характеристика</i>	<i>Вплив на діяльність підприємства</i>
Тип конкуренції - олігополія	Наявність невеликої кількості великих гравців, які мають потужний вплив на ринок	Підвищена увага щодо можливостей конкурентів, готовність відстоювати свою позицію в регулюючих органах
За рівнем конкурентної боротьби - міжнародна	Дуже часто основні замовники - іноземні	Обов'язковий вихід на міжнародний ринок
За галузевою ознакою - міжгалузева	Продукт застосовується у різних галузях	Вимога гнучкості та/або універсальності продукту
Конкуренція за видами товарів - горизонтальна	Високу роздільну здатність ІЧ-зображень зазвичай забезпечують більш якісні камери, а не товари іншого виду	Моніторинг ринку, прагнення замінити існуючі рішення на нові з інших галузей
За характеристиками конкурентних переваг - нецінова	Нові клієнти залучаються вищою якістю продукту	Вдосконалення продукту та створення «ноу-хау» з метою ще й знизити ціну
За інтенсивністю - товарна	Конкуренція між якістю, а не кількістю функцій товару	Підвищення якості продукту.

Висновок: Рішень на базі нейронних мереж на ринку практично нема (зокрема, від найбільших гравців), що дозволить швидше та ефективніше охопити ринок. Як приклад – змінити конкуренцію з горизонтальної до вертикальної, тобто змінити тип товару з апаратно-програмного рішення на програмне.

Таблиця 4.9 – Аналіз конкуренції за М.Портером

	<i>Прямі конкуренти в галузі</i>	<i>Потенційні конкуренти</i>	<i>Клієнти</i>	<i>Товари-замінники</i>
<i>Складові аналізу</i>	FLIR, Infratech, Fluke	NVIDIA, AMD, Adobe	Державний та приватний сектори	Рішення для збільшення роздільної здатності видимих зображень
<i>Висновки</i>	Існують прямі конкуренти, які працюють над збільшенням роздільної здатності ІЧ-зображень	Конкуренти, які займаються видимими зображеннями можуть відносно легко зайти на ринок	Виробник має більший вплив на ринок, аніж клієнт	Див. висновки по потенційних конкурентах

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

<i>№ n/n</i>	<i>Фактор</i>	<i>Обґрунтування</i>
1.	Мультиплатформеність	Більше охоплення ринку
2.	Точність	Більш точне рішення на ринку задовольняє більш вимогливих клієнтів
3.	Швидкодія	Задовольняє клієнтів із великими об'ємами даних для обробки або із підвищеними вимогами до швидкодії
4.	Собівартість	Нижча собівартість приваблює більше клієнтів

Висновок: Аналіз конкурентів визначив, що серед них є багато потужних та потенційно небезпечних гравців, втім із наявними факторами конкурентоспроможності стартап має, щонайменше, можливість швидко зайняти існуючий ринок.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін проєкту

<i>№ n/n</i>	<i>Фактор конкурентоспроможності</i>	<i>Бали 1-20</i>	<i>Рейтинг розроблюваної системи у порівнянні з товарами конкурентів</i>						
			-3	-2	-1	0	+1	+2	+3
1.	Собівартість	18	+						
2.	Точність	19					+		
3.	Швидкість	20	+						

Висновок: За рейтингом сильних та слабких сторін проєкту розроблене рішення перевершує значно конкурентів за швидкістю та собівартістю, однак є не найбільш точним.

Таблиця 4.12 – SWOT-аналіз стартап-проєкту

Сильні сторони: Швидкодія Багатоплатформеність	Слабкі сторони: Залежність від навчаючих вибірок
Можливості: Збільшення бази навчаючих вибірок Удосконалення архітектури	Загрози: Потужні конкуренти

Висновок: SWOT-аналіз відокремив ключові сильні й слабкі сторони проєкту, а також можливості та загрози.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проєкту

<i>№ п/п</i>	<i>Альтернатива ринкової поведінки</i>	<i>Ймовірність отримання ресурсів</i>	<i>Строки реалізації</i>
1.	Рекламна кампанія	Середня	8 місяців
2.	Участь в міжнародних конференціях	Середня	6 місяців
3.	Налагодження прямого контакту із корпоративними клієнтами	Висока	5 місяців

Висновок: Виходячи з результатів попередніх аналізів обрано 3 стратегії ринкового впровадження, однак, через специфіку ринку, найбільш ефективним має бути налагодження прямого контакту із найбільшими корпоративними споживачами.

4.4 Розробка ринкової та маркетингової стратегії продукту

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

<i>№ п/п</i>	<i>Опис профілю цільової групи потенційних клієнтів</i>	<i>Готовність споживачів сприйняти продукт</i>	<i>Орієнтовний попит в межах цільової групи</i>	<i>Інтенсивність конкуренції в сегменті</i>	<i>Простота входу у сегмент</i>
1.	Контроль виробництва	+	+	Середня	+
2.	Системи безпеки та спостереження	+	+	Середня	+

Висновок: Контроль виробництва та рішення для спостереження та безпеки обрано як основні цільові групи через наявний попит та середню конкуренцію.

Таблиця 4.15 – Визначення базової стратегії розвитку

<i>№ п/ п</i>	<i>Обрана альтернатива розвитку проєкту</i>	<i>Стратегія охоплення ринку</i>	<i>Ключові конкурентоспроможні позиції до обраної альтернативи</i>	<i>Базова стратегія розвитку</i>
1.	Прямий контакт із корпоративним и споживачами	Концентрова- ний маркетинг	Виробництво більш якісного продукту	Спеціалізація

Висновок: Обрано спеціалізацію як базову стратегію розвитку через доволі вузькі кола споживачів. Через цей фактор концентрований маркетинг стає головною стратегією охоплення ринку.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

<i>№ n/n</i>	<i>Чи є проект «першопроходцем» на ринку?</i>	<i>Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?</i>	<i>Чи буде компанія копіювати основні характеристики товару конкурента і які?</i>	<i>Стратегія конкурентної поведінки</i>
1.	Не є «першопроходцем»	І те, і інше - так	Математичні методи у прямих конкурентів, архітектуру нейромережі – у потенційних	Стратегія інновацій

Висновок: Проект не є «першопроходцем» у чистому вигляді, однак його особливості дозволяють обрати стратегію інновацій, що має допомогти отримати нових клієнтів.

Таблиця 4.17 – Визначення стратегії позиціонування

<i>№ n/n</i>	<i>Вимоги до товару цільової аудиторії</i>	<i>Базова стратегія розвитку</i>	<i>Ключові конкуренто- спроможні позиції власного стартап- проєкту</i>	<i>Вибір асоціацій, які мають сформувавши комплексну позицію власного проєкту (три ключових)</i>
1.	Швидкодія	Спеціалізація	Швидкодія	Висока швидкість роботи
2.	Точність реконструкції	Спеціалізація	Точність реконструкції	Висока точність реконструкції

Висновок: Швидкодія та точність реконструкції є ключовими в формуванні позиції стартап-проєкту.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

<i>№ n/n</i>	<i>Потреба</i>	<i>Вигода, яку пропонує товар</i>	<i>Ключові переваги перед конкурентами</i>
1.	Швидкодія	Висока швидкодія	Мультиплатформеність, швидкодія, точність, ціна
2.	Точність реконструкції	Точність реконструкції на рівні найкращих рішень	

Висновок: За аналізом конкурентоспроможності товару можна формувати маркетингову концепцію.

Таблиця 4.19 – Опис трьох рівнів моделі товару

<i>Рівні товару</i>		<i>Сутність та складові</i>		
Товар за задумом		Система збільшення роздільної здатності ІЧ-зображень на базі штучних нейронних мереж.		
Товар у реальному виконанні		<i>Властивості/характеристики</i>	<i>М/Нм</i>	<i>Вр/Тх/Тл/Е/Ор</i>
		Швидкодія	М	Тх
		Мультиплатформеність	М	Тх
		Точність		
		Якість: відповідає нормам ISO/IEC 25010:2014 «Інженерія систем та програмного забезпечення – Вимоги до якості систем та програмного забезпечення»		
		Пакування: програмний файл		
Товар із підкріпленням		До продажу – програмне забезпечення		
		Після продажу – сервісна підтримка та оновлення		

Висновок: сформовано модель товару, що дозволяє перехід до формування ціни та збуту.

Таблиця 4.20 – Визначення меж встановлення ціни

<i>№ n/p</i>	<i>Рівень цін на товари-замінники</i>	<i>Рівень цін на товари-аналоги</i>	<i>Рівень доходів цільової групи споживачів</i>	<i>Верхня та нижня межі встановлення ціни на товар/послугу</i>
1.	700-10000 у.о.	0-800 у.о.	2000-20000 у.о.	0-700 у.о.

Висновок: Серед подібних рішень на ринку є як умовно безкоштовні, так і професійні рішення. При формуванні ціни варто розглянути знижену ціну для наукових інституцій та/або вищих навчальних закладів.

Таблиця 4.21 – Формування системи збуту

<i>№ n/n</i>	<i>Специфіка закупівельної поведінки цільових клієнтів</i>	<i>Функції збуту, які має виконувати постачальник</i>	<i>Глибина каналу збуту</i>	<i>Оптимальна система збуту</i>
1.	Продаж ліцензій замовникам	Зв'язок та підтримка замовників	Нульового/ першого рівня	Пряма, збут через сайт виробника

Висновок: Як найбільш зручний, обрано єдиний шлях збуту – напряду через сайт розробника.

Таблиця 4.22 – Концепція маркетингових комунікацій

<i>№ n/n</i>	<i>Специфіка поведінки цільових клієнтів</i>	<i>Канали комунікацій, якими користуються цільові клієнти</i>	<i>Ключові позиції, обрані для позиціону- вання</i>	<i>Завдання рекламного повідомлення</i>	<i>Концепція рекламного звернення</i>
1.	Бажання мати більш точне уявлення про виробничий процес, що спостерігається	Професійні соцмережі, ЗМІ, e-mail, конференції, виставки, рекомендації	Швидкість роботи, точність реконструкції, мультиплатформеність	Донести якомога більше інформації про привабливі сторони продукту	Апробація на відомих наукових конференціях та виставках, створення сайту
2.	Бажання мати більш точні системи спостереження				

Висновок: Професійні канали комунікації обрано як пріоритетні, оскільки там присутня найбільша кількість компаній-споживачів.

Висновок по розділу 4

Аналіз ринку, конкурентів та їхніх пропозицій дають підстави стверджувати, що розроблене рішення окрім наукової цінності, має ще й можливість розвитку у повноцінний стартап-проект із такими особливостями:

- має застосування у сфері контролю виробничих процесів та сфері безпеки;
- споживачі можуть бути як приватними, так і державними підприємствами та службами;
- швидкодія, мультиплатформенність та точність є основними перевагами у боротьбі з конкурентами;
- оскільки запропоноване рішення є програмним, значно спрощується його доставка до кінцевого споживача та існує можливість зручного для клієнта оновлення та підтримки;
- через доволі специфічні сфери застосування, пошук нових клієнтів здійснюється по професійних канал комунікації.

ЗАГАЛЬНІ ВИСНОВКИ

В даній дисертаційній роботі вирішено важливу та актуальну науково-прикладну задачу підвищення роздільної здатності інфрачервоних зображень шляхом удосконалення існуючих методів на основі згорткових нейромереж та створення алгоритмічних і програмних рішень їх реалізації. При цьому отримано наступні наукові результати. Проведено дослідження існуючих методів збільшення роздільної здатності та встановлено:

- Більшість комерційних рішень для підвищення роздільної здатності інфрачервоних зображень використовує декілька інфрачервоних кадрів, з яких потім математичними методами відновлюється зображення із високою роздільною здатністю. Шляхи отримання ІЧ-кадрів можуть бути різними (від застосування дзеркального колеса, що зміщує зображення, до встановлення одночасно кількох ІЧ-сенсорів).
- Оскільки, за великим рахунком, традиційні методи суперроздільної здатності змішують сусідні пікселі із різним співвідношенням для отримання більшого зображення, втрачається високочастотна складова зображення. Але використання нейронних мереж здатне подолати цю проблему.
- Переважна більшість нейронних мереж для збільшення роздільної здатності розроблена та перевірена лише на зображеннях з видимого спектру.
- Покращення якості відновлених зображень нейронними мережами досягалось, в основному, збільшенням кількості згорткових шарів. Однак це має негативні наслідки – ризик перенавчання та збільшення кількості параметрів.
- Додавання рекурсії збільшує глибину мережі без збільшення кількості її параметрів.

Через високу якість збільшення роздільної здатності зображень та відносно малу кількість параметрів обрано BCLSR як основну нейронну

мережу для досліджень та вдосконалення. В роботі вперше досліджено її застосування для збільшення роздільної здатності FIR-зображень, а також запропоновано власну нейронну мережу, що є модифікацією BCLSR. Запропонована мережа відрізняється від BCLSR відсутністю рекурсивних і рекурентних елементів, а також використанням Inception-блоків, що в результаті дозволило збільшити швидкість роботи майже в два рази та навіть дещо покращити якість відновлених зображень на 0,063 дБ. Процедура навчання при цьому виконувалась в один цикл із одноразовим зростанням та спаданням параметру швидкості навчання на нічних FIR-зображеннях навчальної вибірки CVC-14 із однаковими параметрами як для запропонованої, так і для BCLSR мережі.

Незважаючи на те, що запропоноване рішення має більшу кількість параметрів (2,7 М) порівняно з BCLSR (0,6 М), обидві мережі за розміром можуть вважатися малими і використовуватись як у додатках для звичайних персональних комп'ютерів, так навіть і у вбудованих рішеннях.

Можливими напрямками подальших досліджень є вдосконалення архітектури запропонованої мережі шляхом додавання нових типів шарів та (або) зміни гіперпараметрів поточних шарів. Покращення якості роботи також може бути досягнуте переглядом процедури навчання із застосуванням інших функцій втрат та стратегій зміни параметру швидкості навчання.

Попередній аналіз ринку, його гравців та пропозицій показав, що стартап-проект, створений на базі розробленого рішення, є конкурентоспроможним. Також визначено цільову аудиторію продукту, канали комунікації з нею та способи збуту.

Згідно із викладеним, можна зробити висновок, мету дисертації досягнуто, а її завдання – виконано в повному обсязі.

Список використаних джерел

1. Mandanici, E., Tavasci, L., Corsini, F., та ін. A multi-image super-resolution algorithm applied to thermal imagery. // *Applied Geomatics*. - 2019. Vol. 11, No. 3. С. 215–228.
2. Wang, Z., Chen, J., Hoi, S. C. H. Deep Learning for Image Super-resolution: A Survey // *arXiv.org*. Дата оновлення: 08.02.2020. URL: <https://arxiv.org/abs/1902.06068> (Дата звернення: 13.05.2021).
3. UltraMax® – the Ultimate Resolution. URL: <https://www.flir.com/discover/rd-science/ultramax--the-ultimate-resolution/> (Дата звернення: 13.05.21).
4. MicroScan significantly increases the geometrical resolution capability of the ImageIR® camera series. URL: [/search/docs.php3?id=21250&msgID=0&rootID=0](https://www.flir.com/search/docs.php3?id=21250&msgID=0&rootID=0) (Дата звернення: 13.05.21).
5. Multi-Spectral Dynamic Imaging | FLIR Systems. URL: <https://www.flir.com/instruments/multi-spectral-dynamic-imaging/> (Дата звернення: 13.05.21).
6. Liu, S., Yang, Y., Li, Q., та ін. Infrared Image Super Resolution Using GAN With Infrared Image Prior: *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, July.19. С. 1004–1009.
7. Dong, C., Loy, C. C., He, K., та ін. Image Super-Resolution Using Deep Convolutional Networks // *arXiv.org*. Дата оновлення: 31.07.2015. URL: <https://arxiv.org/abs/1501.00092> (Дата звернення: 13.05.2021).
8. Kim, J., Lee, J. K., Lee, K. M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks // *arXiv.org*. Дата оновлення: 11.11.2016. URL: <https://arxiv.org/abs/1511.04587> (Дата звернення: 13.05.2021).
9. Shi, W., Caballero, J., Huszár, F., та ін. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network // *arXiv.org*. Дата оновлення: 23.09.2016. URL: <https://arxiv.org/abs/1609.05158> (Дата звернення: 13.05.2021).

10. Dong, C., Loy, C. C., Tang, X. Accelerating the Super-Resolution Convolutional Neural Network // arXiv.org. Дата оновлення 01.08.2016. URL: <https://arxiv.org/abs/1608.00367> (Дата звернення: 13.05.2021).
11. Ledig, C., Theis, L., Huszar, F., та ін. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network // arXiv.org. Дата оновлення 25.05.2017. URL: <https://arxiv.org/abs/1609.04802> (Дата звернення: 13.05.2021).
12. Lai, W.-S., Huang, J.-B., Ahuja, N., та ін. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution // arXiv.org. Дата оновлення 09.10.2017. URL: <https://arxiv.org/abs/1704.03915> (Дата звернення: 13.05.2021).
13. The Laplacian Pyramid: URL: http://sepwww.stanford.edu/data/media/public/sep/morgan/texturematch/paper_html/node3.html (Дата звернення: 13.05.21).
14. Zhang, K., Zuo, W., Zhang, L. Learning a Single Convolutional Super-Resolution Network for Multiple Degradations // arXiv.org. Дата оновлення 24.05.2018. URL: <https://arxiv.org/abs/1712.06116> (Дата звернення: 13.05.21).
15. Wang, X., Yu, K., Dong, C., та ін. Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform // arXiv.org. Дата оновлення 09.04.2018. URL: <https://arxiv.org/abs/1804.02815> (Дата звернення: 13.05.21).
16. Choi, Y., Kim, N., Hwang, S., та ін. Thermal Image Enhancement using Convolutional Neural Network: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October.16. С. 223–230.
17. Rivadeneira, R. E., Suárez, P. L., Sappa, A. D., та ін. Thermal Image SuperResolution Through Deep Convolutional Neural Network: *Image Analysis and Recognition*, Cham , Springer International Publishing, 19. С. 417–426.

18. Lee, K., Lee, J., Lee, J., та ін. Brightness-Based Convolutional Neural Network for Thermal Image Enhancement. *IEEE Access*. 2017. Vol. 5. C. 26867–26879.
19. Chang, Y., Luo, B. Bidirectional Convolutional LSTM Neural Network for Remote Sensing Image Super-Resolution. *Remote Sensing*. 2019. Vol. 11, No. 20. C. 2333.
20. He, K., Zhang, X., Ren, S., та ін. Deep Residual Learning for Image Recognition // *arXiv.org*. Дата оновлення 10.12.2015. URL: <https://arxiv.org/abs/1512.03385> (Дата звернення: 13.05.21).
21. Sajjadi, M. S. M., Vemulapalli, R., Brown, M. Frame-Recurrent Video Super-Resolution // *arXiv.org*. Дата оновлення 25.03.2018. URL: <https://arxiv.org/abs/1801.04590> (Дата звернення: 13.05.21).
22. Huang, Y., Wang, W., Wang, L. Bidirectional recurrent convolutional networks for multi-frame super-resolution: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, Cambridge, MA, USA , MIT Press, 07.December.15. C. 235–243.
23. Huang, Y., Wang, W., Wang, L. Video Super-Resolution via Bidirectional Recurrent Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2018. Vol. 40, No. 4. C. 1015–1028.
24. Guo, J., Chao, H. Building an end-to-end spatial-temporal convolutional network for video super-resolution: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA , AAAI Press, 04.February.17. C. 4053–4060.
25. Han, W., Chang, S., Liu, D., та ін. Image Super-Resolution via Dual-State Recurrent Networks // *arXiv.org*. Дата оновлення 07.05.2018. URL: <https://arxiv.org/abs/1805.02704> (Дата звернення: 13.05.21).
26. Cars Overhead With Context Dataset at LLNL. URL: <http://gdo-datasci.ucllnl.org/cowc/> (Дата звернення: 13.05.21).

27. Tong, T., Li, G., Liu, X., та ін. Image Super-Resolution Using Dense Skip Connections: *2017 IEEE International Conference on Computer Vision (ICCV)*, October.17. С. 4809–4817.
28. Graves, A., Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM networks: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, July.05. С. 2047–2052 vol. 4.
29. Yaroshenko, M. O., Varfolomieiev, A. Y., Yaganov, P. O. Інерційна система розпізнавання жестів. *Мікросистеми, Електроніка та Акустика*. 2019. Vol. 24, No. 5. С. 42–47.
30. Shi, X., Chen, Z., Wang, H., та ін. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting // arXiv.org. Дата оновлення 19.09.2015. URL: <https://arxiv.org/abs/1506.04214> (Дата звернення: 13.05.21).
31. Wojna, Z., Ferrari, V., Guadarrama, S., та ін. The Devil is in the Decoder: Classification, Regression and GANs // arXiv.org. Дата оновлення 19.02.2019. URL: <https://arxiv.org/abs/1707.05847> (Дата звернення: 13.05.21).
32. Szegedy, C., Ioffe, S., Vanhoucke, V., та ін. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning // arXiv.org. Дата оновлення 23.08.2016. URL: <https://arxiv.org/abs/1602.07261> (Дата звернення: 13.05.21).
33. Bicubic interpolation. Wikipedia: URL: https://en.wikipedia.org/wiki/Bicubic_interpolation (Дата звернення 13.05.2021).
34. Цифровая обработка изображений в информационных системах: Учебное пособие. / [И. С. Грузман, В. С. Киричук, В. П. Косих та ін.]. – Новосибирск: НГТУ, 2002. – 352 с.
35. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981. Vol. 29, No. 6. С. 1153–1160.

36. CVC-14: Visible-FIR Day-Night Pedestrian Sequence Dataset – Elektra. URL: <http://adas.cvc.uab.es/elektra/enigma-portfolio/cvc-14-visible-fir-day-night-pedestrian-sequence-dataset/> (Дата звернення 13.05.2021).
37. Smith, L. N. A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay // arXiv.org. Дата оновлення 24.04.2018. URL: <https://arxiv.org/abs/1803.09820> (Дата звернення 13.05.2021).

Інерційна система розпізнавання жестів

Ярошенко[†] М. О., ORCID [0000-0002-3092-3856](https://orcid.org/0000-0002-3092-3856)

Варфоломєєв[†] А. Ю., к.т.н., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Яганов[†] П. О., к.т.н., доц., ORCID [0000-0001-7358-9846](https://orcid.org/0000-0001-7358-9846)

Кафедра конструювання електронно-обчислювальної апаратури
Факультет електроніки
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Київ, Україна

DOI: [10.20535/2523-4455.2019.24.5.193295](https://doi.org/10.20535/2523-4455.2019.24.5.193295)

Анотація.—В роботі розглянуто інерційну систему розпізнавання жестів. Дана система побудована на основі апаратного модуля, що формує образи-жести за допомогою інерційних MEMS давачів (акселерометра і гіроскопа) та розпізнає отримані образи за допомогою нейронної мережі з BLSTM шаром. Реалізовано прототип запропонованої системи, в якому апаратна частина побудована на мікроконтролері STM32F401RE та мікросхемі MEMS давачів MPU9250, а розпізнавання жестів виконується на персональному комп'ютері засобами пакету прикладних програм Matlab. За допомогою розробленого прототипу сформовано навчальну та тестову вибірки для трьох типів жестів та показано спроможність підсистеми розпізнавання забезпечити похибку ідентифікації даних жестів не більше 3,5% (коефіцієнт правильного розпізнавання 96,7%).

Ключові слова — розпізнавання жестів; інерційна система; нейронні мережі; BLSTM

1. ВСТУП

Розвиток інформаційно-обчислювальних засобів електронних систем створює нові можливості для розв'язку складних ресурсомістких задач, які знаходять своє місце у сучасних системах штучного інтелекту. Серед них чільне місце займають системи розпізнавання образів. При цьому термін «образ» трактується не тільки у розумінні візуального образу, але і як деякий вектор, що однозначно характеризує стан об'єкта.

До розпізнавання образів належать також системи розпізнавання жестів людини. Ці системи знаходять своє застосування для комунікативного спілкування у різних сферах людської діяльності: від спорту і медицини до промисловості і бізнесу. Особливо це актуально для людей з обмеженими можливостями. Згідно даних Всесвітньої організації охорони здоров'я майже 5% населення планети мають проблеми зі слухом, а за даними Всесвітньої федерації глухих у світі налічують понад 70 млн. людей, для яких формою повсякденного спілкування є мова жестів [1].

Системи розпізнавання жестів активно впроваджуються у ігрові консолі відеоігор з віртуальною реальністю, персональні комп'ютери і смартфони [2]. Наприклад, смартфони серії Google Pixel 4 на додаток до системи розпізнавання обличчя Face Unlock мають здатність розпізнавати прості жести завдяки системі Motion Sense.

Розробка ефективних недорогих швидкодіючих систем з розпізнавання жестів триває, а їх інтеграція до складу систем зі штучним інтелектом є актуальною задачею.

Більшість сучасних систем розпізнавання жестів аналізують їх візуальний образ. Проблеми, які при цьому доводиться вирішувати, пов'язані з необхідністю виділити на загальному зображенні об'єкт, який ідентифікують як «жест». Потім цей образ потрібно класифікувати як інформативну складову. При цьому рівень освітленості, нахил світлотіні, колір світла, антропометричні особливості руки людини, що передає жест, положення руки тощо належать до факторів інформаційного шуму, який потрібно фільтрувати. Крім того, в системах реального часу слід визначити початок і кінець жесту у потоці відеокадрів. Для вирішення всіх перелічених вище складностей виникає потреба використання спеціальних алгоритмів обробки даних.

Переважає більшість цих алгоритмів базуються на методах і технологіях нейронних мереж. Оскільки нейронні мережі здатні ефективно здійснювати класифікацію та ідентифікацію образів, вони використовуються як інструмент вирішення різноманітних задач: від відносно простих одношарових перцептронних класифікаторів [3] до складних ідентифікаторів на глибоких нейронних мережах. В роботах [4, 5] повідомляється про використання згорткової нейронної мережі (CNN, англ. «Convolutional Neural Networks»), за допомогою якої здійснюють ідентифікацію жесту на загальному фоні зображення. Створюють також синтез алгоритмів обробки даних CNN, рекурентної нейронної мережі (RNN, англ. «Recurrent Neural Networks») та нейронної мережі з довгою короткостроковою пам'яттю (LSTM, англ. «Long Short-Term Memory») [6]. Точність ідентифікації жестів в таких системах зазвичай знаходиться в межах від 77% до 97% [2, 6]. Структури нейронних мереж при цьому можуть бути доволі складними, а кількість глибоких



Copyright (c) 2019 Ярошенко М. О., Варфоломєєв А. Ю., Яганов П. О.

шарів досягати десятків [7, 8]. Все це вимагає значних обчислювальних ресурсів, що можуть бути забезпечені лише сучасними надпотужними мікропроцесорами та графічними прискорювачами.

Водночас, для зменшення обчислювального навантаження на систему розпізнавання жестів, замість обробки візуальних образів, використовують спеціальні електронно-механічні пристрої, що реєструють рух за сигналами з датчиків, створюючи попередній вектор-образ жесту, який потім ідентифікує нейронна мережа. Конструктивно такі пристрої виконані у формі рукавичок, що надіваються на руку. Датчиками сигналів зазвичай слугують або мікромеханічні інерційні пристрої (гіроскопи, акселерометри), або гнучкі тензорезистори, які змінюють свій опір під час деформації суглобів пальців (виробник Spectra Symbol Flex Sensor [9]). Такі апаратно-програмні рішення є менш вартісними у порівнянні з оптичними системами, не містять складних блоків попередньої обробки даних та потребують значно менших обчислювальних ресурсів. З огляду на це, в даній роботі вирішено зосередитись на дослідженні систем саме такого типу.

Таким чином, метою роботи є встановлення особливостей функціонування системи розпізнавання жестів на рекурентній BLSTM нейронній мережі з інерційними MEMS датчиками. Поставлена мета досягається за рахунок вирішення наступних задач:

- розроблення апаратного пристрою для формування образу-жесту на основі інерційних MEMS датчиків;
- створення процедури розпізнавання поточних даних на основі алгоритму функціонування нейронної мережі з довгою короткостроковою пам'яттю (LSTM).

Дослідження системи розпізнавання виконуватиметься на прикладі трьох жестів: рух по колу (circle), постукування (tap) та невідомий жест (other).

II. АПАРАТНИЙ БЛОК ЗАХОПЛЕННЯ РУХУ

Дані для розпізнавання жестів отримані за допомогою апаратного пристрою для захоплення руху, структурна схема якого показана на рис. 1, а.

Макет пристрою реалізовано засобами налагоджувальних плат NUCLEO-F401RE на базі мікроконтролера STM32F401RE, модуля GY-9250 з мікросхемою MPU9250, що включає трьохосові акселерометр і гіроскоп, а також модуля-перехідника Troyka Shield для забезпечення зв'язку між ними (рис. 1, б). Макет пристрою працює наступним чином. Мікроконтролер здійснює налаштування режимів роботи блоку датчиків та отримує від них дані шиною I²C. Далі він пересилає отримані дані через віртуальний COM-порт на ПК з частотою 25 Гц, де в середовищі MATLAB виконується розпізнавання жестів.

Для зменшення впливу шумів використовуються внутрішні цифрові фільтри сенсора: 92 Гц частота зрізу для гіроскопа та 99 Гц – для акселерометра. Діапазон вимірювань акселерометра $\pm 2g$, гіроскопа – ± 250 °/с.

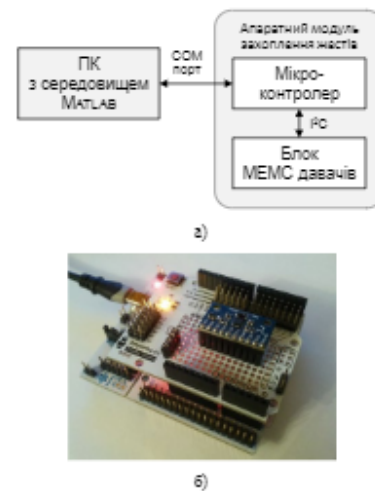


Рис. 1 Пристрій для захоплення руху: а) структурна схема пристрою; б) макет пристрою

III. ЗАГАЛЬНА СТРУКТУРА НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ЖЕСТІВ

Для розпізнавання жестів використовується нейронна мережа з BLSTM шаром (цей шар також інколи називають BiLSTM) [10], яка належить до класу рекурентних. Ця нейромережа є модифікованою версією більш простої мережі LSTM, тому наведемо спершу короткі відомості саме про неї.

Нейромережі LSTM мають властивість «запам'ятовувати» вектори-образи за рахунок існування у їх структурі зворотних зв'язків. У багатословних нейронних рекурентних нейромережах з'являється можливість охопити такими зв'язками не лише вихідний шар із вхідним, але і внутрішні шари у різних комбінаціях. Внаслідок цього у таких нейромережах з'являється можливість не тільки запам'ятовувати образ, але і накопичувати інформацію, здійснюючи на цій основі прогностичну функцію.

Особливість функціонування LSTM нейронної мережі полягає у тому, що синаптичні коефіцієнти петлі зворотнього зв'язку у ній не постійні, а програмно керовані. Цим досягають посилення ефекту самонавчання, оскільки процес накопичення інформації для прийняття рішень стає динамічним і залежить від вхідної послідовності векторів-образів [11]. Крім того, LSTM нейромережа має здатність «забувати» непотрібні складові інформації під час обробки послідовності даних. Це забезпечують спеціальні блоки-вентилю, що входять до її складу. Особливість їх функціонування ілюструє рис. 2.

Стан c_t комірки LSTM у момент часу t визначається наступним чином:

$$c_t = f_t \otimes c_{t-1} + g_t \otimes i_t,$$

а її прихований стан h_t , як:



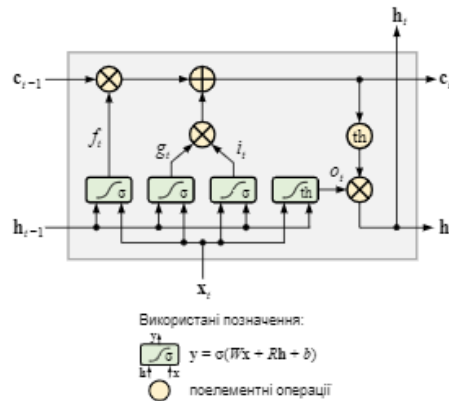


Рис. 2 Комірка LSTM

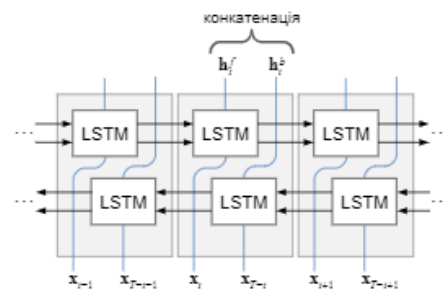


Рис. 3 Розгортка шару BLSTM нейронної мережі

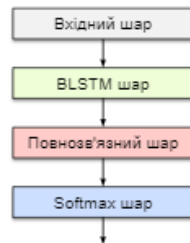


Рис. 4 Структура нейронної мережі для розпізнавання жестів із BLSTM шаром

$$h_t = o_t \otimes \text{th}(c_t),$$

де \otimes – позначає поелементне множення; $\text{th}(\cdot)$ також застосовується поелементно.

Значення f_t у комірку формується вентилем забування і контролює ступінь видалення старої інформації зі стану комірки. Значення g_t формується вентилем кандидата і визначає інформацію, що має додаватися до стану комірки. Значення i_t формується вентилем входу та задає рівень оновлення стану комірки. Значення o_t обчислюється вентилем виходу та контролює рівень інформації, що потрапляє зі стану комірки в її

прихований стан. Наведені значення обчислюються наступним чином:

$$\begin{aligned} f_t &= \sigma(W_f x_t + R_f h_{t-1} + b_f), \\ i_t &= \sigma(W_i x_t + R_i h_{t-1} + b_i), \\ g_t &= \sigma(W_g x_t + R_g h_{t-1} + b_g), \\ o_t &= \text{th}(W_o x_t + R_o h_{t-1} + b_o) \end{aligned} \quad (1)$$

де W – вагові коефіцієнти входу; R – рекурентні вагові коефіцієнти та b – зміщення; x_t – вектор вхідного сигналу в поточний момент часу; h_{t-1} – прихований стан комірки в попередній момент часу; $\sigma(x) = (1 + e^{-x})^{-1}$ – сигмоїдна функція активації. Добутки у виразах (1) є матричними або скалярними – залежно від розмірів x_t , h_{t-1} , W , R та b .

Розміри вхідного вектора і матриць W , R та b визначають кількість елементів векторів стану c_t та прихованого стану h_t комірки. Кількість елементів у векторі h_t дорівнює кількості прихованих елементів LSTM шару.

Шар BLSTM мережі будується на основі LSTM комірок. На рис. 3 показано розгортку даного шару. По суті він включає дві LSTM комірки, що дозволяє краще виявляти взаємозв'язки між вимірюваннями у послідовності в двох напрямках – від початку до кінця послідовності та навпаки. Цей тип шару показав свою ефективність при розв'язанні таких задач аналізу послідовностей, як класифікація фону, розпізнавання тексту, розпізнавання ритму та інших. Також, BLSTM шар у випадку розпізнавання жестів є менш чутливим до довжини послідовності жесту за часом та зсуву жесту в часі [11].

Навчання LSTM та BLSTM шарів виконується тими ж методами, що і звичайних рекурентних мереж. Зокрема, одним з найбільш поширених способів є алгоритм *зворотного розповсюдження помилки в часі* (BPTT) [10, 11], який передбачає розгортання рекурентного шару у послідовність шарів без зворотного зв'язку, до яких далі застосовується звичайний алгоритм зворотного розповсюдження помилки.

Повна нейронна мережа, що виконує розпізнавання вхідної послідовності, має вигляд, показаний на рис. 4.

У нейронній мережі, показаній на рис. 4, вхідний шар виконує функцію поелементного передавання послідовності вхідного сигналу до наступного шару. Вхідний сигнал при цьому складається зі 100 нормованих до діапазону $[-1, +1]$ вимірювань акселерометра та гіроскопа по трьом осям, тобто по 6 значень на кожне вимірювання. Вимірювання здійснюються з частотою вибірки 25 Гц. Зазначений розмір послідовності та частота вимірювань обрані за рекомендаціями з [12]. Із вхідного шару дані послідовно передаються до шару BLSTM, який формує вихідний вектор з вдвічі більшою кількістю елементів, ніж число його прихованих вузлів, що обумовлено наявністю двох LSTM комірок у даному шарі. Після обробки всіх вимірювань вхідної послідовності вихід BLSTM шару передається на повнозв'язний шар з трьома нейронами, які класифікують отриману послідовність відповідно на 3 класи. Остаточна класифікація вхідної



послідовності виконується шаром з м'яким максимумом (Softmax шаром), на який передаються сигнали з попереднього повнозв'язного шару. Для повноти викладення наведемо формулу, за якою обчислюється значення функції Softmax:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}},$$

де x_i та y_i – це відповідно i -й вхід та вихід шару; j – пробігає індекси всіх N виходів шару (в мережі, що розглядається $N=3$).

Під час навчання мережі використовується функція втрат на основі взаємної ентропії, що мінімізує похибку мережі з вихідним Softmax-шаром:

$$L = \sum_{i=1}^N \sum_{j=1}^K t_{ij} \ln(y_{ij}),$$

де N – кількість послідовностей у навчальній вибірці; K – кількість класів; t_{ij} – індикатор належності i -ї послідовності навчальної вибірки j -му класу; y_{ij} – значення j -го виходу Softmax-шару для i -ї послідовності навчальної вибірки.

IV. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

Для того, щоб виконувати розпізнавання жестів, застосовуючи описану в попередньому розділі нейронну мережу, її необхідно навчити. Крім того, необхідно також впевнитись, що навчена нейронна мережа здатна розпізнавати жести належним чином. Для цього за допомогою пристрою захоплення руху було отримано навчальну та тестову вибірки. Зокрема, навчальна вибірка складається з 228 послідовностей (по 76 послідовностей на кожний клас жестів) розміром 6×100 (100 вимірювань гіроскопа та акселерометра по кожній з 3-х осей). Тестова вибірка має аналогічну структуру, проте складається з 30 послідовностей (по 10 прикладів на кожен жест).

Для налаштування параметрів нейронної мережі було випробувано декілька методів навчання: стохастичний градієнтний спуск з інерційністю (SGDM), RMSProp та Adam [11]. Проте за досягнутим значенням коефіцієнта розпізнавання найкращим серед них виявився метод з адаптивною інерційністю Adam [13], в якому використовувались наступні параметри:

- швидкість навчання: $\alpha = 0,01$;
- коефіцієнт затухання градієнта: $\beta_1 = 0,99$;
- коефіцієнт затухання квадрата градієнта: $\beta_2 = 0,99$;
- параметр для запобігання ділення на 0: $\epsilon = 10^{-8}$;
- максимальна кількість епох: $N_E = 2000$;
- розмір навчальної підмножини (mini-batch) $B = 30$.

ТАБЛИЦЯ 1 Залежність коефіцієнту розпізнавання від кількості прихованих елементів у BLSTM шарі

Кількість прихованих елементів у BLSTM шарі	Точність розпізнавання, %
25	83,3
50	86,7
75	86,7
100	96,7
125	86,7

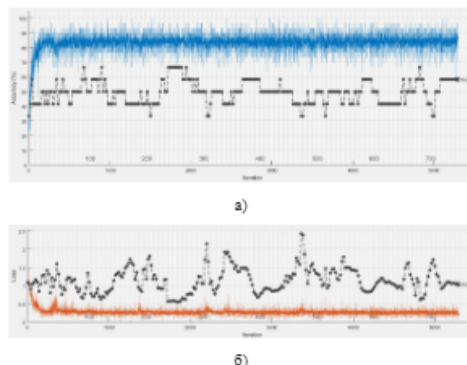


Рис. 5 Динаміка навчання нейронної мережі: а) зміна точності розпізнавання (синій колір, суцільна лінія); б) зміна значення функції втрат (червоний колір, суцільна лінія)

Під час експериментів було досліджено нейронні мережі з різною кількістю прихованих елементів у BLSTM шарі (див. табл. 1). Виявлено, що найкращий результат розпізнавання досягається при 100 прихованих елементах в BLSTM шарі. Цікаво відмітити, що при використанні 50, 75 та 125 прихованих елементів у BLSTM шарі, точність розпізнавання є однаковою і меншою та складає 86,7%. Причиною цього можуть бути особливості початкової ініціалізації ваг нейронів під час навчання, що призводить до потрапляння алгоритму оптимізації у локальний мінімум функції втрат.

Динаміка зміни точності розпізнавання під час навчання (співпадіння результатів роботи нейронної мережі із навчальною вибіркою) та зміни значення функції втрат наведена на рис 5. Зауважимо, що задовільна якість розпізнавання досягається вже при 756 епохах навчання.

Випробовування нейронної мережі на тестовій вибірці, приклади з якої не приймали участь під час навчання, показало вірність розпізнавання жестів на рівні 96,7%, що може вважатися цілком прийнятним результатом.

Висновки

В роботі запропоновано та досліджено апаратно-програмний комплекс розпізнавання жестів. Встановлено, що апаратна реалізація блоку формування сигналів жестів можлива на доступній елементній базі – мікроконтролері STM32 та модуля MEMS-давачів GY-9250, а класифікацію жестів може здійснювати рекурентна BLSTM нейромережа, синтезована та навчена засобами пакету прикладних програм



МАТЛАВ. Процедура розпізнавання потокових даних на основі алгоритму функціонування нейронної мережі BLSTM при цьому здатна забезпечити похибку ідентифікації жестів в межах до 3,5%, що є прийнятним для подібного класу систем. Це обґрунтовує ефективність процесу розпізнавання жестів апаратно-програмними засобами, представленими у даній роботі, оскільки корисний результат досягається незначними затратами, які зумовили його одержання.

Експериментально визначено оптимальну кількість прихованих елементів у BLSTM шарі нейронної мережі заданої конфігурації, що опосередковано вказує на особливості функціонування системи, для якої характерна нелінійна залежність між кількістю прихованих елементів і якістю розпізнавання жестів.

Подальші дослідження можуть бути спрямовані на підвищення ідентифікаційних можливостей системи з розпізнавання жестів за рахунок підвищення роздільної здатності давачів апаратної частини та якості навчання BLSTM нейромережі. Крім того, може бути підвищена й автономність пристрою шляхом перенесення на мікроконтролер обчислень виходів нейронної мережі та передавання розпізнаного жесту через бездротовий канал.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] "World Federation of the Deaf," [Online]. Available: <http://wfd deaf.org/faq>. [Accessed 22 08 2019].
- [2] S. Alashhab, A.-J. Gallego and M. Lozano, "Hand gesture detection with convolutional neural networks," *Advances in Intelligent Systems and Computing*, p. 45–52, 2018. DOI: 10.1007/978-3-319-94649-8_6.
- [3] P. O. Yahanov and I. V. Redko, "Perseptronnyi klasyfikator teplovoho komfortu," *Viznyk KNUITD*, no. 6 (128), p. 29–38, 2018. DOI: 10.30857/1813-6796.2018.6.3.
- [4] J. Tompson, M. Stein, Y. LeCun and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, p. 169–173, 2014. DOI: 10.1145/2629500.
- [5] S. Ji, W. Xu, M. Yang and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, vol. 35, no. 1, p. 221–231, 2013. DOI: 10.1109/TPAMI.2012.59.
- [6] N. Nishida and H. Nakayama, "Multimodal gesture recognition using multi-stream recurrent neural network," *Image and Video Technology PSIVT 2015. Lecture Notes in Computer Science*, vol. 9431, p. 682–694, 2015. DOI: 10.1007/978-3-319-29451-3_54.
- [7] N. Neverova, C. Wolf, G. W. Taylor and F. Nebout, "Multi-scale deep learning for gesture detection and localization," *Computer Vision – ECCV 2014 Workshops. ECCV. Lecture Notes in Computer Science*, vol. 8925, p. 474–490, 2014. DOI: 10.1007/978-3-319-16178-5_33.
- [8] R. V. Siriak and I. S. Skarha-Bandurova, "Model obrobky potokovykh danykh dlia rozpoznavannia okremykh odyntys zhestovoi movy," *Viznyk Natsionalnoho tekhnichnoho universytetu "KhPI"*, vol. 42, no. 1318, p. 73–81, 2018.
- [9] "Spectra Symbol Flex Sensor," [Online]. Available: <https://www.spectrasymbol.com/product/flex-sensors/>. [Accessed 22 08 2019].
- [10] A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," *Neural Networks*, vol. 18, no. 5–6, p. 602–610, 2005. DOI: 10.1016/j.neunet.2005.06.042.
- [11] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, Massachusetts: MIT Press, 2016. ISBN: 978-0262035613.
- [12] G. Lefebvre, S. Berlemont, F. Mamalet and C. Garcia, "Inertial Gesture Recognition with BLSTM-RNN," in *Artificial Neural Networks: Methods and Applications in Bio-Neuroinformatics*, P. Koprinkova-Hristova, V. Mladenov and K. N. K., Eds., Springer International Publishing Switzerland, 2015, pp. 393–410. DOI: 10.1007/978-3-319-09903-3_19.
- [13] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, San Diego, 2015. URL: <https://arxiv.org/pdf/1412.6980.pdf>.

Надійшла до редакції 20 вересня 2019 р.

УДК 004.358

Инерционная система распознавания жестов

Ярошенко[†] М. А., ORCID [0000-0002-3092-3856](https://orcid.org/0000-0002-3092-3856)

Варфоломеев[‡] А. Ю., к.т.н., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Яганов[§] П. А., к.т.н., доц., ORCID [0000-0001-7358-9846](https://orcid.org/0000-0001-7358-9846)

Кафедра конструирования электронно-вычислительной аппаратуры
Факультет электроники
Национальный технический университет Украины
"Киевский политехнический институт имени Игоря Сикорского"
Киев, Украина

Аннотация. В работе рассмотрены инерциальную систему распознавания жестов. Данная система построена на основе аппаратного модуля, формирующего образы-жесты с помощью инерционных МЭМС датчиков



Copyright (c) 2019 Ярошенко М. О., Варфоломеев А. Ю., Яганов П. О.

(акселерометра и гироскопа) и распознает полученные образы с помощью нейронной сети с BLSTM слоем. Реализовано прототип предложенной системы, в котором аппаратная часть построена на микроконтроллере STM32F401RE и микросхеме МЭМС давай MPU9250, а распознавание жестов выполняется на персональном компьютере средствами пакета прикладных программ MATLAB. С помощью разработанного прототипа сформирован учебную и тестовую выборки для трех типов жестов и показано способность подсистемы распознавания обеспечить погрешность идентификации данных жестов не более 3,5% (коэффициент правильного распознавания 96,7%).

Ключевые слова — распознавание жестов; инерционная система; нейронные сети; BLSTM

UDC 004.358

The Inertial Gesture Recognition System

M. A. Yaroshenko[†], ORCID [0000-0002-3092-3856](https://orcid.org/0000-0002-3092-3856)

A. Yu. Varfolomeiev[‡], PhD, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

P. A. Yaganov[§], PhD Assoc.Prof., ORCID [0000-0001-7358-9846](https://orcid.org/0000-0001-7358-9846)

Department of Design of Electronic Computing Equipment

Faculty of Electronics

National technical university of Ukraine "Igor Sikorsky Kyiv polytechnic institute"

Kyiv, Ukraine

Abstract—The paper is devoted to the implementation of the gestures recognition system. Existing gesture recognition approaches are considered. Among them the system of inertial type is chosen for research, since the systems of this type are less difficult to implement and require relatively low computational complexity. Thus, the proposed system is based on two components: the hardware module for gesture capturing and the subsystem for the recognition of captured gestures. The hardware component of the system employs MEMS accelerometer and gyroscope for motion acquisition. The prototype of this component is implemented using the STM32F401RE microcontroller and the MEMS sensor MPU9250. The communication between the microcontroller and the MEMS sensor is carried out via the I2C bus. The microcontroller initializes the sensor and acquires the motion data. Then it sends the collected data via the virtual COM port to the PC, where the captured motions are recognized. The system component for the recognition of gestures is implemented as the neural network in MATLAB environment. This network consists of four layers: the input layer, the BLSTM layer, the fully-connected layer, and the Softmax layer. The key component of the selected neural network architecture is the BLSTM layer, which (due to its properties) provides some invariance to the length and the shift of the gesture signal in time. The suggested system was tested on three gesture classes: circular motion, tapping and other/unknown gesture. Using the hardware module the training and the testing sets were collected. In course of experiments, it was established that the best training results for the selected network architecture and the collected training set are achieved using the Adam optimizer. The experiment with different number of hidden units in the BLSTM layer showed that the highest recognition rate may be achieved when 100 units are used. Particularly, the suggested system with the specified number of hidden units in BLSTM layer was able to attain the recognition error less than 3.5% (recognition rate of 96.7%) on the test set.

Keywords — gesture recognition; inertial system; neural networks; BLSTM



Copyright (c) 2019 Ярошенко М. О., Варфоломеев А. Ю., Яганов П. О.

Ієрархічна згорткова нейронна мережа для підвищення роздільної здатності інфрачервоних зображень

Ярошенко[†] М. О., ORCID [0000-0002-3092-3856](https://orcid.org/0000-0002-3092-3856)

Варфоломєєв^{*} А. Ю., к.т.н., доц., ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)

Яганов^{*} П. О., к.т.н., доц., ORCID [0000-0001-7358-9846](https://orcid.org/0000-0001-7358-9846)

Кафедра конструювання електронно-обчислювальної апаратури, Факультет електроніки
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Київ, Україна

Анотація — Через відносно високу ціну інфрачервоних сенсорів, існує необхідність у методах обробки інфрачервоних зображень, отриманих із недорогих ІЧ-камер, які б підвищували їхню роздільну здатність. Одним з напрямків таких методів є надроздільна здатність (SR – Super Resolution) на основі згорткових нейронних мереж (CNN – Convolutional Neural Network), що активно розвивається останні десятиріччя. Однак більшість існуючих нейронних SR-мереж створено для збільшення роздільної здатності зображень у видимому спектрі. В цьому ж дослідженні на інфрачервоних зображеннях перевірено працездатність однієї з кращих мереж для збільшення роздільної здатності зображень у видимому спектрі – BCLSR, та запропоновано власну мережу, яка є швидшою та точніше відновлює (збільшує) роздільну здатність вхідних зображень.

Ключові слова — підвищення роздільної здатності зображень; згорткові нейронні мережі; рекурентні нейронні мережі; інфрачервоні зображення

1. Вступ

В останні десятиліття застосування тепловізорів розширюється завдяки експлуатації наземних, бортових та космічних платформ. Наземні спостереження можуть використовуватися, наприклад, для виявлення втрат теплової енергії будівлями або для моніторингу робочих умов протікання промислових виробничих процесів, тоді як аерофотозйомка застосовується для підтримки міського планування і точного землеробства і т.д. Усі зазначені застосування потребують точної і докладної інформації про температуру поверхонь та їх просторову структуру.

Незважаючи на значний розвиток технологій виробництва теплових сенсорів, викликаний зростаючим попитом, вони досі володіють низькою роздільною здатністю, яка є помітно нижчою в порівнянні з зображеннями, отриманими у видимій смузі або в ближній інфрачервоній області. Виключаючи спеціалізоване супутникове обладнання, більшість наявних на ринку інфрачервоних камер мають роздільну здатність менше одного мегапикселя. З апаратної точки зору подальше збільшення роздільної здатності ускладнене тим, що або зменшення розміру пікселя призведе до зниження відношення сигнал/шум, або збільшення їхньої кількості призведе до збільшення

розміру сенсора, що було б занадто дорого для пристроїв загального призначення.

У цьому контексті надроздільна здатність, що являє собою процес генерування зображення з більш високою роздільною здатністю (HR – High Resolution) на основі з одного або декількох кадрів нижчої роздільної здатності (LR – Low Resolution), є перспективним підходом для подолання поточних технологічних обмежень і забезпечення можливості аналізу дрібних деталей температурних характеристик поверхонь [1].

Провідні виробники тепловізорів почали впроваджувати алгоритми надроздільної здатності в програмне забезпечення своїх останніх моделей. Наприклад, компанія FLIR розробила технологію UltraMax, яка заснована на швидкому отриманні 16 теплових кадрів менш ніж за 1 с [2]. Через природні рухи оператора, кожне зображення буде дещо зміщене щодо інших. В результаті виходить нове зображення з подвоєною роздільною здатністю за умови, що користувач або ціль майже нерухомі [1]. Замість цього Infratech використовує апаратне рішення, що складається з швидкого обертового колеса MicroScan [3], яке дозволяє отримувати чотири експозиції, кожне з яких зміщено в поперечному напрямку на половину пікселя. Інші програмні рішення засновані на об'єднанні даних із зображеннями у видимій смузі. FLIR MSX [4],



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

наприклад, підвищує чіткість зображення шляхом нанесення деталей зображення видимого спектру на теплові зображення. Однак такий алгоритм, імовірно, спотворить теплове вимірювання, тому що більш дрібні деталі беруться з видимої смуги і не обов'язково відповідають змінам температури поверхонь, що спостерігаються [1].

Принципово іншим є підхід із застосуванням нейронних мереж, який ґрунтується на наявності попереднього «знання» алгоритмом типових форм об'єктів у зображенні низької роздільної здатності [5]. У 2015 році була вперше [6] застосована згорткова нейронна мережа (SRCNN – Super Resolution Convolutional Neural Network) для реконструкції зображення. SRCNN є згортковою нейронною мережею для відновлення високої роздільної здатності зображень, яка навчається на великій кількості пар зображень із різною роздільною здатністю. Хоча запропонований у [6] алгоритм і є кращим за звичайну бікубічну інтерполяцію, тришарова структура SRCNN все ж не дозволяє досягти більш високої якості зображень, оскільки мала кількість шарів дає лише «поверхневу» інформацію про вміст зображення. Щоб отримати більш детальні характеристики, а отже і кращий результат реконструкції, можна спробувати використовувати більш глибоку мережу, що власне і було запропоновано в методі VDSR [7], який має вже 20 згорткових шарів [5].

Зображення з низькою роздільною здатністю у вищезазначених підходах спочатку бікубічно інтерполюються, а потім оброблюються нейронною мережею. Для досягнення продуктивності в реальному часі було запропоновано мережу ESPCN [8], яка безпосередньо обробляє зображення з низькою роздільною здатністю і забезпечує її підвищення за рахунок збільшення числа каналів характеристичних ознак (feature channels) [5].

SRGAN [9] використовує GAN (змагальна породжуюча мережа) для поліпшення деталей зображення. SRGAN, хоч і має значення PSNR (Peak Signal-to-Noise Ratio – пікове співвідношення сигнал-шум) нижче за ESPCN, але згенеровані нею зображення мають вищий показник MOS (Mean Opinion Score), тобто, на думку незалежних оцінювачів, суб'єктивно виглядають більш якісними та близьким до оригіналу.

Робота [10] є однією з перших, де розглядається питання підвищення роздільної здатності теплових зображень за допомогою згорткових нейронних мереж. В цій роботі використовуються ідеї з праці [6], а її автори порівнюють точність мережі, навченої на зображеннях з різних спектральних діапазонів, щоб отримати краще уявлення про можливості застосування згорткових нейронних мереж для підвищення роздільної здатності теплових зображень. Вони, зокрема, прийшли до висновку, що мережа, навчена на відтинках сірого, забезпечує кращий результат масштабування, ніж навчена на безпосередньо теплових зображеннях MWIR-спектру (Mid-wavelength infrared) [11].

В роботі [12] запропоновано згорткову нейронну мережу для реконструкції теплових зображень, що навчається методом залишкового навчання (residual learning). При цьому автори роботи, з огляду на обмеженість навчальних вибірок з тепловими зображеннями, використовують для навчання звичайні кольорові зображення, представлені у спеціальних просторах кольору та оцінюють ефективність використання для навчання одної з кольорових компонент даних просторів, а саме: відтинки сірого, яскравість L в просторі HSL, інтенсивність I в просторі HSI та компонент V з кольорового простору HSV. Автори приходять до висновку, що навчання на V компоненті дає найкращий результат відновлення, а запропонований ними підхід також ще й покращує швидкість навчання [11].

Слід відмітити, що в роботі [11], на відміну від [10] та [12], запропоновано нейромережу, навчену виключно на (FIR – far infrared) теплових зображеннях, яка показала кращі результати, ніж дві попередні архітектури.

Згадані до цього моменту архітектури нейронних мереж складаються в основному зі згорткових шарів. Втім, глибока модель CNN (створена шляхом додавання більшої кількості згорткових шарів) призводить до суттєвого збільшення кількості параметрів та ускладнює процедуру навчання. Збільшення глибини нейронних мереж підвищує також і ризик перенавчання [13]. Для боротьби із цим існує 3 основні підходи (Рис.1):

- пропуск шарів (Рис.1, а): як глобальне залишкове з'єднання (residual connection), яке пов'язує вхідні дані і вихідний шар, як, наприклад, в [7, 11, 12], так і локальне залишкове з'єднання з ResNet [14], зменшують труднощі в навчанні;
- структури з щільними зв'язками (dense connection structures) пов'язують всі шари в мережі і об'єднують карти об'єктів всіх попередніх рівнів (Рис.1, б), щоб зменшити зникаючі градієнти і повторно використовувати елементи з неглибоких шарів (shallow layers);
- рекурсивні нейронні (Рис.1, в) мережі багаторазово застосовують одні й ті ж згорткові шари для керування параметрами з досягненням великих рецептивних полів [13].

У задачі надроздільної здатності рекурсивні нейронні мережі зазвичай використовуються для захоплення довгострокової залежності сусідніх кадрів у відео [15]. Однак, в роботі [13] запропоновано нейронну мережу BCLSR (Рис.2), що складається з двох структур – набору блоків рекурсивного виводу (RIB) та двонапрявленого згорткового LSTM шару (BiConvLSTM), що оброблюють послідовність, утворену з'єднаними виходами блоків рекурсивного виводу. Тобто результат кожної з ітерацій рекурсивного блоку є елементом послідовності для рекурсивного LSTM шару.



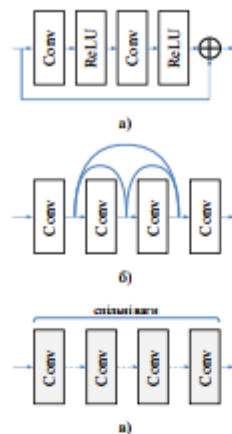


Рис. 1. Основні підходи до зменшення глибини нейронної мережі: а) пропуск шарів; б) структури з цільними зв'язками; в) рекурсивні нейронні мережі (сірим позначено рекурсивні згорткові шари)

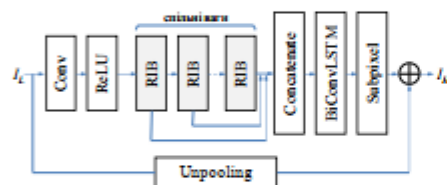


Рис. 2. Архітектура мережі BCLSR

Автори BCLSR стверджують, що запропонована ними мережа перевершує аналоги за рахунок таких аспектів:

- 1) Через збільшення поля сприйняття з кожною рекурсією, більше контекстної інформації може використовуватись для відновлення якісного зображення без додаткових параметрів.
- 2) Завдяки повторному використанню низькорівневих ознак, потік інформації через мережу може бути підсилений без «вибуху» або «зниження» градієнту.
- 3) Двонаправлений згортковий LSTM шар вибірково витягує доповнюючу інформацію з усіх рекурсій та запобігає надходженню зайвої інформації до шару відновлення.

Таким чином, нейромережі на базі LSTM шарів на даний момент є кращими за якістю відновлення високої роздільної здатності, складністю навчання та кількістю параметрів. В літературних джерелах не виявлено жодної згадки про застосування LSTM шарів до обробки інфрачервоних зображень, що і спонукало присвятити дане дослідження в тому числі використанню архітектури BCLSR для

підвищення роздільної здатності інфрачервоних зображень. Окрім цього в даній роботі було запропоновано власну архітектуру мережі без застосування рекурсивних шарів та порівняно якість її роботи з мережею BCLSR.

Таким чином розробка власної архітектури нейронної мережі та порівняння її з однією з найкращих мереж для підвищення роздільної здатності зображень і складає основну мету даного дослідження.

II. BiConvLSTM МЕРЕЖА ДЛЯ ПІДВИЩЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ІНФРАЧЕРВОНИХ ЗОБРАЖЕНЬ

Завдяки описаним в минулому розділі перевагам BCLSR мережі, її було обрано за основу запропонованого методу підвищення роздільної здатності інфрачервоних зображень з метою перевірки її архітектурних особливостей на FIR-зображеннях та можливості удосконалення або адаптації до цього класу зображень. Для повноти викладення розглянемо основні елементи архітектури BCLSR більш детально.

A. Блок рекурсивного виводу (RIB)

Як зазначалося раніше, одним з основних елементів мережі є блок рекурсивного виводу (RIB – Recursive Inference Block) – розроблений під впливом архітектури іншої мережі ResNet [14], що застосовується для класифікації зображень. Структура даного блоку показана на Рис. 3.



Рис. 3. Блок рекурсивного виводу (RIB) мережі BCLSR. F_{i-1} та F_i позначають вихід попередньої ітерації та вхід наступної ітерації відповідно

В цілому, використання компонентів із мереж для класифікації у мережах, що призначені для підвищення роздільної здатності, є виправданим, оскільки припускається, що в обох випадках використовуються одні й ті самі ознаки вхідних зображень. Однак, результат їх виділення є різним: відгуки згорткових шарів під час класифікації збуджують нейрони, що відповідають різним класам, тоді як в задачах підвищення роздільної здатності вони є основою для додавання деталей до зображення.

Б. Двонаправлені LSTM шари (BiConvLSTM)

Окремо варто розглянути роботу двонаправлених LSTM-шарів. Оригінальні LSTM-шари запропоновано в 1997 в [16] з метою збереження інформації про довготривалі залежності між послідовними даними. В BCLSR використано згорткові LSTM комірки [17], що мають наступні рівняння виходів та станів:



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

$$\begin{aligned}
 i_t &= \sigma(x_t * w_{xi} + h_{t-1} * w_{hi} + b_i), \\
 f_t &= \sigma(x_t * w_{xf} + h_{t-1} * w_{hf} + b_f), \\
 o_t &= \sigma(x_t * w_{xo} + h_{t-1} * w_{ho} + b_o), \\
 c_t &= f_t \cdot c_{t-1} + i_t \cdot \tanh(x_t * w_{xc} + h_{t-1} * w_{hc} + b_c), \\
 h_t &= o_t \cdot \tanh(c_t),
 \end{aligned} \quad (1)$$

де $\sigma(\cdot)$ позначає сигмоїдну функцію; $*$ позначає згортку; \cdot позначає по-елементне множення; x_t – поточний (t -й) елемент оброблюваної послідовності; w_{mn} – ваги фільтру для згортки з сигналу m для розрахунку вентилу n ($m = \{x, h\}$, $n = \{i, f, o, c\}$); b_n – зміщення для вентилу n ($n = \{i, f, o, c\}$); f_t – вентиль «забування», що визначає яку частину інформації з попередньої ітерації відкинути при обробці поточного елемента послідовності; i_t – вхідний вентиль; o_t – вихідний вентиль; c_t – стан комірки, що відповідає за збереження довготривалих залежностей при обробці послідовностей; h_t – вихід комірки.

У BCLSR використовується два BiConvLSTM шари, кожен з яких містить по дві згорткові LSTM комірки, через які послідовності сигналів пропускаються в прямому та зворотному напрямках [13].

B. Depth-to-Space шар (Subpixel)

Для узгодження за розмірами, перед додаванням до інтерполюваного вхідного зображення, вихід попереднього шару спочатку оброблюється згорткою, щоб отримати тензор розміром $[h_{LK}, w_{LK}, c \times f_{upscale}^2]$, а потім об'єднується в один тензор, що має розмір $[h_{LK} \times f_{upscale}, w_{LK} \times f_{upscale}, 1]$ ($f_{upscale}$ – коефіцієнт масштабування) як показано на Рис. 4 [18].

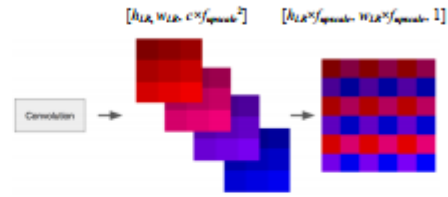


Рис. 4. Приклад операції Depth-to-Space

III. ЗАПРОПОНОВАНА МЕРЕЖА

BCLSR за даними, наведеними в оригінальній статті [13], дійсно перевершує попередні відомі нейронні мережі для підвищення роздільної здатності, однак вона має деякі недоліки: по-перше, застосування рекурсивного блоку хоч і зменшує кількість параметрів для навчання та дозволяє регулювати глибину нейронної мережі, втім, таке рішення може бути менш гнучким, ніж застосування нейронної мережі із такою самою глибиною, але різними параметрами шарів, по-друге, схожий недолік має й рекурентний LSTM шар – не дивлячись на те, що при розрахунку кожної ітерації поточне вхідне значення та значення минулої ітерації «комбінуються» в різних пропорціях, протягом обробки вхідної послідовності значення ваг не змінюються зі збільшенням глибини, що також впливає на гнучкість нейронної мережі.

Для подолання вказаних недоліків було розроблено власну мережу, архітектура якої наведена на Рис. 5.

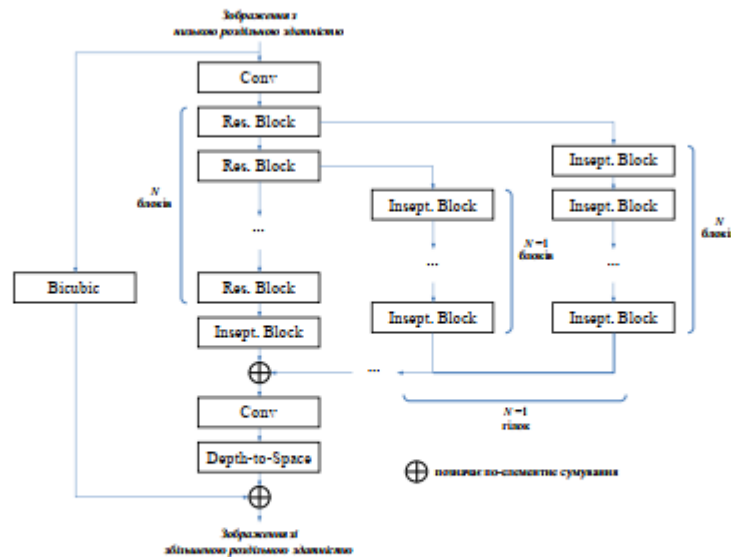


Рис. 5. Архітектура запропонованої мережі



Суть її роботи дуже схожа на BCLSR – вхідні дані послідовно оброблюються однаковими за структурою блоками шарів, після чого результати їх роботи комбінуються задля отримання кінцевого зображення з великою роздільною здатністю. Відмінності запропонованої архітектури:

- 1) відмова від LSTM-шарів із їхньою заміною на «гілки» з кількох однакових блоків;
- 2) застосування RIB з BCLSR як структурного блоку, але не рекурсивно.

Архітектура має деревоподібну (ієрархічну) структуру, а вихід кожного із послідовних Residual блоків (структура даного блоку аналогічна RIB блоку, наведеному на Рис. 3, на Рис. 5 позначений як Res.block) утворює гілку, що складається із послідовно з'єднаних Inception-блоків (на Рис. 5 позначений як Incept.block), структура яких розглядається нижче. Виходи усіх гілок сумуються і до отриманої суми застосовується перетворення Depth-to-Space. Наступним і останнім кроком є додавання результату останньої операції до вхідного зображення, розширеного за допомогою бікубічної інтерполяції.

Для подальшої обробки ознак, вилучених після кожного Residual-блоку, як вже зазначалося раніше, створюється окрема «гілка». Кожна така «гілка» складається із послідовно з'єднаних однакових за структурою блоків (Рис. 6), створених на основі однієї з найкращих нейронних мереж для розпізнавання зображень Inception [19], та, відповідно, позначених як Incept.block.

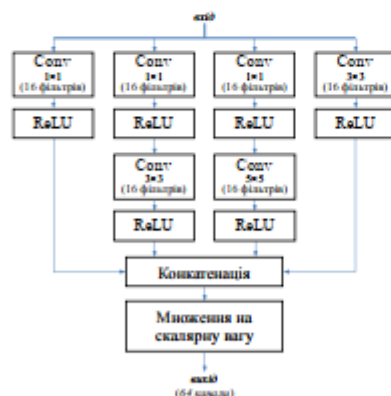


Рис. 6. Запропонований Inception-блок

Після кожного із згорткових шарів в Inception блоках додано dropout шари. За умови невеликої кількості прикладів у навчальних вибірках з FIR зображеннями, додавання dropout шару має запобігти перенавчанню мережі.

IV. ПРАКТИЧНІ РЕЗУЛЬТАТИ

A. Особливості навчання

Моделювання запропонованої нейронної мережі виконано засобами програмного пакету Tensorflow в середовищі Google Colaboratory. Досліджувався варіант запропонованої мережі із 8 гілками ($N=8$ на Рис.3) яка збільшує роздільну здатність в 2 рази. В усіх подальших порівняннях застосовується BCLSR із 8 застосуваннями рекурсивного блоку, яку також реалізовано та навчено засобами Tensorflow для цього дослідження. Для наближення умов порівняння із запропонованим рішенням, згорткові LSTM-шари також використовують dropout. В RIB та Residual блоках встановлено однакову кількість каналів в згорткових шарах – 64.

Для навчання та тестування запропонованої мережі та BCLSR використано FIR-зображення, зняті вночі, з навчальної вибірки CVC-14 [20]. Дана вибірка не є великою, тому задля її розширення з кожного зображення випадковим чином було вирізано 20 шматків розміром 100×100 , до значень яскравості яких було додано випадкові числа та застосовано випадкову зміну контрастності (функції `random_brightness` та `random_contrast` пакету Tensorflow), до деяких зображень також застосовано відзеркалення. Отримана навчальна вибірка поділяється на підвибірki (mini-batch) розміром по 8 зображень.

Для оптимізації моделі застосовано оптимізатор Adam із $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$. Функція втрат $L = \frac{1}{m} \|f - f_{GT}\|^2$ – середньоквадратична похибка між зображенням, що генерує мережа f та еталонним зображенням f_{GT} з високою роздільною здатністю $m \times n$ пікселів.

У роботі [21] показано доцільність циклічної зміни швидкості навчання (одного з параметрів оптимізатора), подібна політика застосована і під час навчання запропонованої мережі – швидкість навчання зростає з 10^{-4} до 10^{-3} за 10 епох, потім так само спадає, додатково після цього одну епоху вона дорівнює $9.1 \cdot 10^{-6}$. Графік зміни швидкості навчання наведений на Рис. 7.

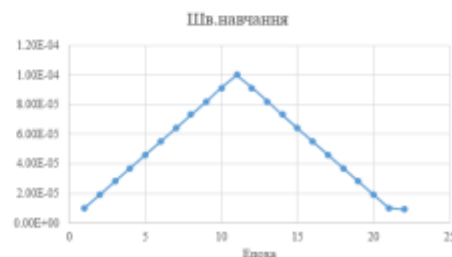


Рис. 7. Графік зміни швидкості навчання

Б. Результати тестування мереж

Після 1 циклу навчання отримано нейронну мережу, що відновлює зображення із високою роздільною здатністю із середнім PSNR на



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

тестовому наборі (FIR-зображення 640×480 пікселів) в 42,2749 дБ, в той час як для бікубічної інтерполяції цей показник дорівнює 40,5721 дБ. Таким чином, маємо приріст PSNR в 1,7028 дБ.

Навчена аналогічним чином мережа BCLSR на тому ж тестовому наборі вічних FIR-знімків CVC-14 забезпечує середній PSNR в 42,2119 дБ. Тобто дає приріст PSNR в 1,6398 дБ порівняно з бікубічною інтерполяцією.

Таким чином, якість зображень відновлених запропонованою мережею є дещо вищою в порівнянні з BCLSR. Візуальні результати відновлення для одного з зображень тестової вибірки CVC-14 показано на Рис. 8.

Запропонована мережа є помітно більшою за BCLSR. Так, зокрема, вона має 2697512 параметрів, тоді як BCLSR – 591940 параметрів. Не зважаючи на це, запропоноване рішення є більш швидкодіючим: час обробки одного зображення на відеокарті NVIDIA Tesla P100-PCIE-16GB для нього становить 0,126 с, хоча для BCLSR це значення дорівнює 0,24 с. Причиною цього є відносна складність операцій в LSTM шарах мережі BCLSR – як вже зазначалося, згортковий LSTM шар включає кілька згорток, по-елементне множення, підсумовування та, що важливо, обчислення сигмодних функцій та гіперболічних тангенсів (1).



Рис. 8. Візуальне порівняння оригінального зображення, його утиснутого варіанту (для покращення сприйняття збільшеного методом найближчого сусіда), відновленого BCLSR, відновленого запропонованою нейронною мережею («Proposed») та відновленого бікубічною інтерполяцією

V. ВИСНОВКИ

В статті вперше досліджено застосування нейронної мережі BCLSR для збільшення роздільної здатності FIR-зображень, а також запропоновано

власну нейронну мережу, що є модифікацією BCLSR. Запропонована мережа відрізняється від BCLSR відсутністю рекурсивних і рекурентних елементів, а також використанням Insertion-блоків, що в результаті дозволило збільшити швидкість роботи майже в два рази та навіть дещо покращити якість відновлених зображень на 0,063 дБ. Процедура навчання при цьому виконувалась в один цикл із одноразовим зростанням та спаданням параметру швидкості навчання на вічних FIR-зображеннях навчальної вибірки CVC-14 із однаковими параметрами як для запропонованої, так і для BCLSR мережі.

Незважаючи на те, що запропоноване рішення має більшу кількість параметрів (2,7 М) порівняно з BCLSR (0,6 М), обидві мережі за розміром можуть вважатися малими і використовуватись як у додатках для звичайних персональних комп'ютерів, так навіть і у вбудованих рішеннях.

Можливими напрямками подальших досліджень є вдосконалення архітектури запропонованої мережі шляхом додавання нових типів шарів та (або) зміни гіперпараметрів поточних шарів. Покращення якості роботи також може бути досягнуте переглядом процедури навчання із застосуванням інших функцій втрат та стратегій зміни параметру швидкості навчання.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] E. Madsen, L. Terasaki, F. Corsini and a. al., "A multi-image super-resolution algorithm applied to thermal imagery," *Appl. Geomat.*, vol. 11, p. 215–228, 2019. DOI: [10.1007/s12518-019-00253-y](https://doi.org/10.1007/s12518-019-00253-y).
- [2] "ULTRAMAX – The Ultimate Resolution." [Online]. Available: https://www.firmmedia.com/DMC/THG/Brochure/IND_023/IN_D_023_EN.pdf. [Accessed 05 03 2021].
- [3] "MicroScan significantly increases the geometrical resolution capability of the ImageIR® camera series," [Online]. Available: <https://www.ndt.net/search/docs.php?id=31230&count=1>. [Accessed 03 05 2021].
- [4] "FLIR MSX Multi-Spectral Dynamic Imaging." [Online]. Available: <https://www.tequipment.net/Flir/MSX/Multi-Spectral/Dynamic/Imaging.html>. [Accessed 03 05 2021].
- [5] S. Liu, Y. Yang, Q. Li, H. Feng, Z. Xu and Y. Chen, "Infrared Image Super Resolution Using GAN With Infrared Image Prior," in *2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP)*, Wuxi, China, 2019. DOI: [10.1109/ICSIP.2019.8868766](https://doi.org/10.1109/ICSIP.2019.8868766).
- [6] C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence (TPAMI)*, vol. 35, no. 2, pp. 295–307, 2014. DOI: [10.1109/TPAMI.2015.2439261](https://doi.org/10.1109/TPAMI.2015.2439261).
- [7] J. Kim, J. K. Lee and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016. DOI: [10.1109/CVPR.2016.182](https://doi.org/10.1109/CVPR.2016.182).
- [8] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken and R. Bishop, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016. DOI: [10.1109/CVPR.2016.207](https://doi.org/10.1109/CVPR.2016.207).



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

- [9] C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. DOI: [10.1109/CVPR.2017.19](https://doi.org/10.1109/CVPR.2017.19).
- [10] Y. Choi, N. Kim, S. Hwang and I. S. Kwon, "Thermal image enhancement using convolutional neural network," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea (South), 2016. DOI: [10.1109/IROS.2016.7779039](https://doi.org/10.1109/IROS.2016.7779039).
- [11] R. E. Rivadeneira, P. L. Suárez, A. D. Sappa and B. X. Vintimilla, "Thermal Image SuperResolution Through Deep Convolutional Neural Network," *Image Analysis and Recognition, ICLAR 2019. Lecture Notes in Computer Science*, vol. 11663, 2019. DOI: [10.1007/978-3-030-27272-2_37](https://doi.org/10.1007/978-3-030-27272-2_37).
- [12] K. Lee, J. Lee, J. Lee, S. Hwang and S. Lee, "Brightness-based convolutional neural network for thermal image enhancement," *IEEE Access*, vol. 5, pp. 26867-26879, 2017. DOI: [10.1109/ACCESS.2017.2769637](https://doi.org/10.1109/ACCESS.2017.2769637).
- [13] Y. Chang and B. Luo, "Bidirectional Convolutional LSTM Neural Network for Remote Sensing Image Super-Resolution," *Remote Sens.*, vol. 11, no. 2333, 2019. DOI: [10.3390/rs11202333](https://doi.org/10.3390/rs11202333).
- [14] K. Ha, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [15] M. S. M. Sajjadi, R. Vanulapalli and M. Brown, "Frame-Recurrent Video Super-Resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018. DOI: [10.1109/CVPR.2018.006693](https://doi.org/10.1109/CVPR.2018.006693).
- [16] A. Graves and J. Schmidhuber, "Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. (3-6), p. 602-610, 2005. DOI: [10.1016/j.neunet.2005.06.047](https://doi.org/10.1016/j.neunet.2005.06.047).
- [17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2015.
- [18] Z. Wojna, V. Ferrari, S. Guadarrama, N. Silberman, L.-C. Chen, A. Fathi and J. Uijlings, "The Devil is in the Decoder: Classification, Regression and GANs," *International Journal of Computer Vision (IJCV)*, vol. 127, p. 1694-1706, 2019. DOI: [10.1007/s11263-019-01170-8](https://doi.org/10.1007/s11263-019-01170-8).
- [19] C. Szegedy, S. Ioffe, V. Vanhoucke and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in *AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [20] "CVC-14: Visible-FIR Day-Night Pedestrian Sequence Dataset," [Online]. Available: <http://das.cvc.uab.es/eletra/enigma-portfolio/cvc-14-visible-fir-day-night-pedestrian-sequence-dataset/>. [Accessed 04 05 2021].
- [21] L. N. Smith, "A disciplined approach to neural network hyperparameters: Part 1 - learning rate, batch size, momentum, and weight decay," *CoRR*, vol. abs/1803.09820, 2018.

UDC 004.932.4

Hierarchical convolutional neural network for infrared image super-resolution

M. A. Yaroshenko[†], ORCID [0000-0002-3092-3856](https://orcid.org/0000-0002-3092-3856)A. Y. Varfolomeiev[‡], PhD, ORCID [0000-0002-6990-7140](https://orcid.org/0000-0002-6990-7140)P. A. Yaganov[‡], PhD, ORCID [0000-0001-7358-9846](https://orcid.org/0000-0001-7358-9846)

Department of Design of Electronic Computing Equipment, Faculty of Electronics

National technical university of Ukraine

"Igor Sikorsky Kyiv polytechnic institute"

Kyiv, Ukraine

Abstract — Due to the high price of thermal imaging sensors, methods for high quality upscaling of infrared images, acquired from low resolution inexpensive IR-cameras become of high demand. One of the very promising branches of such kind of methods is base on super-resolution (SR) techniques that exploit convolutional neural networks (CNN), which are developed rapidly for the last decade. During the review of existing solutions, we found that the most of super-resolution neural networks are intended to the upscaling of images in the visible spectrum band. Among them, the BCLSR network has proven to be one of the best solutions that ensure very high quality of image upscaling. Thus, we selected this network for further investigation in the current paper. Namely, in this research we trained and tested the BCLSR network for upscaling of far infrared (FIR) images for the first time. Moreover, inspired by the BCLSR architecture, we proposed our own neural network, which differs from the BCLSR by the absence of recursive and recurrent layers that were replaced by series connected Residual- and parallel connected Inception-like blocks correspondingly. During the tests, we found that the suggested modifications permit to increase the network inference speed almost twice and even improve the quality of upscaling by 0,063 dB compared to the basic BCLSR implementation. Networks were trained and tested using the CVC-14 dataset that contains FIR-images acquired at the night. We used data augmentation with random dividing



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

dataset images onto 100×100 pixel patches and with subsequent application random brightness, contrast, and mirroring to the obtained patches. The training procedure was performed in a single cycle with single increasing and decreasing of learning rate, and used the same parameters for the proposed and the BCLSR networks. We employed the Adam optimizer for both networks training. Nevertheless, the proposed model has more parameters (2,7 M) compared to the BCLSR (0,6 M), both of the networks can be considered as the small ones, and thus can be used in applications for conventional personal computers, as well as in embedded solutions. The direction of the further research can be focused on the improvements of the proposed network architecture by introducing new types of layers as well as on the modifying of hyperparameters of the used layers. The quality of the upscaling can be increased also by using other loss functions and by the change of learning rate-varying strategies.

Keywords — super-resolution; convolutional neural networks; recurrent networks; far infrared images

DOI: 10.20535/2523-4455.y.v.i.n



Copyright (c) 20XX Прізвище І. П., Прізвище І. П.

Додаток Б. Лістинг програми

Підготовка навчаючої вибірки:

```
current_path = "./drive/MyDrive/datasets/CVC-
14/Night/FIR/NewTest/FramesPos"
save_path = "./pics_for_dataset" # New size of cropped images
NEW_IMG_HEIGHT = 100
NEW_IMG_WIDTH = 100

# ipdb.set_trace()
if os.path.exists(save_path) == False:
    os.mkdir(save_path)

for root, dirs, files in os.walk(current_path, topdown=False):
    for name in files:
        if os.path.splitext(os.path.join(root, name))[1].lower() ==
".jpg":
            # print(os.path.join(root, name))
            save_dir = os.path.join(save_path, os.path.splitext(name)[0])
            os.mkdir(save_dir)
            for i in range(20):
                outputfile = os.path.join(save_dir, os.path.splitext(name)[0]
+ "_" + str(i) + ".jpg")
                try:
                    # ipdb.set_trace()
                    im = Image.open(os.path.join(root, name))
                    im = tf.keras.preprocessing.image.img_to_array(im)
                    im = tf.convert_to_tensor(im)
                    processed_im = tf.image.random_crop(im, size =
[NEW_IMG_HEIGHT, NEW_IMG_WIDTH, 3], seed = 1337)
                    processed_im = tf.image.random_brightness(processed_im,
0.4, seed = 1337)
                    processed_im = tf.image.random_contrast(processed_im, 0.2,
0.8, seed = 1337)
                    processed_im =
tf.image.random_flip_left_right(processed_im, seed = 1337)
                    tf.keras.preprocessing.image.save_img(path=outputfile,
x=processed_im)
                except Exception as e:
                    print(e)
```

Створення датасетів:

```
root_dir = "./pics_for_dataset"

crop_size = NEW_IMG_HEIGHT
upscale_factor = 2
input_size = crop_size // upscale_factor
batch_size = 8
```

```

val_split=0.2

# get size of dataset
files = []
for r, d, f in os.walk(root_dir):
    for file in f:
        if '.jpg' in file:
            files.append(os.path.join(r, file))
dataset_size = round((1 - val_split) * len(files), 0)
print(dataset_size)

```

```

train_ds = image_dataset_from_directory(
    root_dir,
    batch_size=batch_size,
    image_size=(crop_size, crop_size),
    validation_split=val_split,
    subset="training",
    seed=1337,
    label_mode=None,
)

```

```

valid_ds = image_dataset_from_directory(
    root_dir,
    batch_size=batch_size,
    image_size=(crop_size, crop_size),
    validation_split=val_split,
    subset="validation",
    seed=1337,
    label_mode=None,
)

```

Масштабування значень пікселів в інтервал [0;1]:

```

def scaling(input_image):
    input_image = input_image / 255.0
    return input_image

```

```

# Scale from (0, 255) to (0, 1)
train_ds = train_ds.map(scaling)
valid_ds = valid_ds.map(scaling)

```

Підготовка тестової вибірки:

```

import random

test_path = "./drive/MyDrive/datasets/CVC-14/Night/FIR/NewTest/FramesPos"

```

```
test_img_paths = [
    os.path.join(test_path, fname)
    for fname in os.listdir(test_path)
    if fname.endswith(".jpg")
]
```

```
random.shuffle(test_img_paths)
```

Перетворення зображень в навчаючій та валідаційній виборках – зменшення кількості каналів до 1 та зменшення вхідних зображень:

```
# Use TF Ops to process.
def process_input(input, input_size, upscale_factor):
    input = tf.image.rgb_to_yuv(input)
    last_dimension_axis = len(input.shape) - 1
    y, u, v = tf.split(input, 3, axis=last_dimension_axis)
    return tf.image.resize(y, [input_size, input_size], method="area")
```

```
def process_target(input):
    input = tf.image.rgb_to_yuv(input)
    last_dimension_axis = len(input.shape) - 1
    y, u, v = tf.split(input, 3, axis=last_dimension_axis)
    return y
```

```
train_ds = train_ds.map(
    lambda x: (process_input(x, input_size, upscale_factor),
process_target(x))
)
train_ds = train_ds.prefetch(buffer_size=32)
```

```
valid_ds = valid_ds.map(
    lambda x: (process_input(x, input_size, upscale_factor),
process_target(x))
)
valid_ds = valid_ds.prefetch(buffer_size=32)
```

Модель BCLSR:

```
from tensorflow.python.keras import backend as K

def InceptBlock(channels, num_of_dense_layers = 3, activation =
"relu"):
    inputs = keras.Input(shape=(None, None, channels))

    x = keras.layers.ReLU()(inputs)
```

```

    # First branch
    conv1 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(x)
    conv1 = tf.keras.layers.Dropout(0.1)(conv1)

    # Second branch
    conv2 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(x)
    conv2 = tf.keras.layers.Dropout(0.1)(conv2)
    conv2 = tf.keras.layers.Conv2D(channels/4, 3,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(conv2)
    conv2 = tf.keras.layers.Dropout(0.1)(conv2)

    # Third branch
    conv3 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(x)
    conv3 = tf.keras.layers.Dropout(0.1)(conv3)
    conv3 = tf.keras.layers.Conv2D(channels/4, 5,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(conv3)
    conv3 = tf.keras.layers.Dropout(0.1)(conv3)

    # Fourth branch
    # conv4 = tf.keras.layers.MaxPool2D((3,3))(inputs)
    conv4 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation, data_format="channels_last")(x)
    conv4 = tf.keras.layers.Dropout(0.1)(conv4)

    # Concatenaion of branches
    convs = keras.layers.Concatenate()([conv1, conv2, conv3, conv4])
    outputs = keras.layers.Add()([x, convs])

    return keras.Model(inputs, outputs)

def TrueResBlock(channels, num_of_dense_layers = 3, activation =
"relu"):

    inputs = keras.Input(shape=(None, None, channels))
    x = tf.keras.layers.Conv2D(channels, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation)(inputs)

    dense_ins = []
    dense_ins.append(x)
    for i in range(num_of_dense_layers):
        concat_list = []

```

```

    if (len(dense_ins) > 1):
        for j in range(len(dense_ins)):
            concat_list.append(dense_ins[j])
            x = keras.layers.Concatenate()(concat_list)
            x = tf.keras.layers.Conv2D(channels, 3,
kernel_initializer="Orthogonal", padding="same", activation =
activation)(x)
            dense_ins.append(x)

        x = keras.layers.Concatenate()(dense_ins)
        x = tf.keras.layers.Conv2D(channels, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation)(x)
        outputs = tf.keras.layers.Add()([inputs, x])

    return keras.Model(inputs, outputs)

def get_model(upscale_factor=2, channels=1):
    conv_args = {
        "activation": "relu",
        "kernel_initializer": "Orthogonal",
        "padding": "same",
    }

    inputs = keras.Input(shape=(None, None, channels))
    x = layers.Conv2D(64, 3, kernel_initializer="Orthogonal",
padding="same", activation='relu',
data_format="channels_last")(inputs)

    rb = TrueResBlock(64, activation='relu')
    concat_list = []

    for i in range(0, 8):
        block_in = x
        block_out = rb(block_in)
        concat_list.append(block_out)
        x = block_out

    x = tf.stack(concat_list, axis=1)

    x = tf.keras.layers.Bidirectional(tf.keras.layers.ConvLSTM2D(64,
1, kernel_initializer="Orthogonal", padding="same",
return_sequences=True, dropout = 0.1, recurrent_dropout = 0.1,
data_format="channels_last"))(x)

    x = layers.Conv2D(128, 3, kernel_initializer="Orthogonal",
padding="same", activation='relu', data_format="channels_last")(x)
    print(x.shape)
    x = tf.keras.layers.Bidirectional(tf.keras.layers.ConvLSTM2D(64,
1, kernel_initializer="Orthogonal", padding="same",

```

```

return_sequences=True, dropout = 0.1, recurrent_dropout = 0.1,
data_format="channels_last"))(x)

    x = tf.unstack(x, axis=1)
    x = tf.keras.layers.Concatenate(axis=-1)(x)

    x = layers.Conv2D(channels * (upscale_factor ** 2), 3,
kernel_initializer="Orthogonal", padding="same",
data_format="channels_last")(x)
    x = tf.nn.depth_to_space(x, upscale_factor)

    global_res_path =
tf.keras.layers.UpSampling2D(size=(upscale_factor,
upscale_factor))(inputs)
    # global_res_path = ScaleLayer(2)(inputs)
    outputs = tf.keras.layers.Add()([x, global_res_path])

    return keras.Model(inputs, outputs)

model = get_model()

model.summary()

from tensorflow.keras.utils import plot_model
plot_model(model, expand_nested=True)

```

Модель HCNNSR:

```

from tensorflow.python.keras import backend as K

class ScaleLayer(tf.keras.layers.Layer):
    def __init__(self, upscale_factor, **kwargs):
        super(ScaleLayer, self).__init__()
        self.upscale_factor = upscale_factor

    def call(self, inputs):
        if (K.int_shape(inputs)[1] == None) or (K.int_shape(inputs)[2]
==None):
            return inputs
        else:
            return tf.image.resize(inputs, [K.int_shape(inputs)[1] *
self.upscale_factor, K.int_shape(inputs)[2] * self.upscale_factor],
method='bicubic')

    def get_config(self):
        config = {
            'upscale_factor':self.upscale_factor
        }

```

```

        base_config = super(ScaleLayer, self).get_config()
        return dict(list(base_config.items()) + list(config.items()))

class CoeffLayer(tf.keras.layers.Layer):
    def __init__(self, **kwargs):
        super(CoeffLayer, self).__init__()

    def build(self, input_shape):
        self.coeff = self.add_weight("coeff", shape = ())

    def call(self, inputs):
        if (K.int_shape(inputs)[1] == None) or (K.int_shape(inputs)[2]
        ==None) or (K.int_shape(inputs)[3] ==None):
            return inputs
        else:
            return tf.math.scalar_mul(self.coeff, inputs)

    def get_config(self):
        base_config = super(CoeffLayer, self).get_config()
        return dict(list(base_config.items()))

def InceptBlock(channels, num_of_dense_layers = 3, activation =
"relu"):
    inputs = keras.Input(shape=(None, None, channels))
    # First branch
    conv1 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(inputs)
    conv1 = tf.keras.layers.Dropout(0.1)(conv1)

    # Second branch
    conv2 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(inputs)
    conv2 = tf.keras.layers.Dropout(0.1)(conv2)
    conv2 = tf.keras.layers.Conv2D(channels/4, 3,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(conv2)
    conv2 = tf.keras.layers.Dropout(0.1)(conv2)

    # Third branch
    conv3 = tf.keras.layers.Conv2D(channels/4, 1,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(inputs)
    conv3 = tf.keras.layers.Dropout(0.1)(conv3)
    conv3 = tf.keras.layers.Conv2D(channels/4, 5,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(conv3)
    conv3 = tf.keras.layers.Dropout(0.1)(conv3)

```

```

    # Fourth branch
    # conv4 = tf.keras.layers.MaxPool2D((3,3))(inputs)
    conv4 = tf.keras.layers.Conv2D(channels/4, 3,
kernel_initializer="GlorotNormal", padding="same",
activation=activation)(inputs)
    conv4 = tf.keras.layers.Dropout(0.1)(conv4)

    # Concatenaion of branches
    outputs = keras.layers.Concatenate(axis = -1)([conv1, conv2, conv3,
conv4])

    return keras.Model(inputs, outputs)

def ResBlock(channels, num_of_dense_layers = 3, activation = "relu"):

    inputs = keras.Input(shape=(None, None, channels))
    x = tf.keras.layers.Conv2D(channels, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation)(inputs)

    dense_ins = []
    dense_ins.append(x)
    for i in range(num_of_dense_layers):
        concat_list = []
        if (len(dense_ins) > 1):
            for j in range(len(dense_ins)):
                concat_list.append(dense_ins[j])
            x = keras.layers.Concatenate()(concat_list)
            x = tf.keras.layers.Conv2D(channels, 3,
kernel_initializer="Orthogonal", padding="same", activation =
activation)(x)
            dense_ins.append(x)

        x = keras.layers.Concatenate()(dense_ins)
        x = tf.keras.layers.Conv2D(channels, 1,
kernel_initializer="Orthogonal", padding="same",
activation=activation)(x)
        outputs = tf.keras.layers.Add()([inputs, x])

    return keras.Model(inputs, outputs)

def get_model(upscale_factor=2, channels=1):
    inputs = keras.Input(shape=(None, None, channels))
    x = tf.keras.layers.Conv2D(64, 1, kernel_initializer="GlorotNormal",
padding="same", activation="relu")(inputs)

    starts = []

    for i in range(8):
        x = ResBlock(64)(x)

```



```

        starts.append(x)

    branches = []
    for i in range(8):
        branch = starts[i]
        for j in range(8-i):
            # branch = tf.keras.layers.Conv2D(64, 1,
kernel_initializer="GlorotNormal", padding="same",
activation="relu")(branch)
            branch = InceptBlock(64)(branch)
            branch = CoeffLayer()(branch)
            # branch = tf.keras.layers.Conv2D(64, 1,
kernel_initializer="GlorotNormal", padding="same")(branch)
            branches.append(branch)

    x = keras.layers.Add()(branches)

    x = layers.Conv2D(channels * (upscale_factor ** 2), 3,
kernel_initializer="Orthogonal", padding="same")(x)
    x = tf.nn.depth_to_space(x, upscale_factor)
    # global_res_path =
tf.keras.layers.UpSampling2D(size=(upscale_factor, upscale_factor),
interpolation="bilinear")(inputs)
    # ipdb.set_trace()
    global_res_path = ScaleLayer(2)(inputs)
    print(global_res_path.shape)
    outputs = tf.keras.layers.Add()([x, global_res_path])
    return keras.Model(inputs, outputs)

model = get_model()

model.summary()

from tensorflow.keras.utils import plot_model
plot_model(model, expand_nested=False)

```

Імплементація допоміжних функцій (підготовка вхідного зображення, застосування нейронної мережі, відображення результатів):

```

import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1.inset_locator import zoomed_inset_axes
from mpl_toolkits.axes_grid1.inset_locator import mark_inset
import PIL
import time

def plot_results(img, prefix, title):
    """Plot the result with zoom-in area."""
    img_array = img_to_array(img)
    img_array = img_array.astype("float32") / 255.0

```

```

# Create a new figure with a default 111 subplot.
fig, ax = plt.subplots()
im = ax.imshow(img_array[::-1], origin="lower")
# plt.axis('off')

plt.title(title)
# zoom-factor: 2.0, location: upper-left
axins = zoomed_inset_axes(ax, 2, loc=1)
axins.imshow(img_array[::-1], origin="lower")

# Specify the limits.
x1, x2, y1, y2 = 150, 250, 300, 400
# Apply the x-limits.
axins.set_xlim(x1, x2)
# Apply the y-limits.
axins.set_ylim(y1, y2)

plt.yticks(visible=False)
plt.xticks(visible=False)

# Make the line.
mark_inset(ax, axins, loc1=1, loc2=3, fc="none", ec="black")
plt.savefig(str(prefix) + "-" + title +
".png",bbox_inches='tight')
plt.show()

def get_lowres_image(img, upscale_factor):
    """Return low-resolution image to use as model input."""
    return img.resize(
        (img.size[0] // upscale_factor, img.size[1] //
upscale_factor),
        PIL.Image.BICUBIC,
    )

def upscale_image(model, img):
    """Predict the result based on input image and restore the image
as RGB."""
    ycbcr = img.convert("YCbCr")
    y, cb, cr = ycbcr.split()
    y = img_to_array(y)
    y = y.astype("float32") / 255.0

    input = np.expand_dims(y, axis=0)
    start = time.perf_counter()
    out = model.predict(input)
    elapsed = time.perf_counter() - start
    print('Elapsed %.4f seconds.' % elapsed)
    out_img = out * 255.0

```

```

out_img_y = out[0]
out_img_y *= 255.0

# # Restore the image in RGB color space.
out_img_y = out_img_y.clip(0, 255)

out_img_y = tf.convert_to_tensor(out_img_y)
out_img_y = tf.image.grayscale_to_rgb(out_img_y)

return out_img_y

```

Формування параметрів навчання:

```

# Form piecewise schedule according to One cycle recommendations
min_lr = 1e-5
max_lr = 1e-4

# min_lr = 1e-3
# max_lr = 1e-2

cycle_epochs = 20
extra_epochs = 2
val_step = (max_lr - min_lr) / (cycle_epochs // 2)
boundaries = []
values = []
# increase LR
for i in range(cycle_epochs // 2):
    values.append(min_lr + i * val_step)
    boundaries.append(i+1)
# decrease LR
for i in range(cycle_epochs // 2):
    values.append(max_lr - i * val_step)
    boundaries.append(cycle_epochs // 2 + i + 1)

# last epochs with extra low LR
for i in range(extra_epochs):
    values.append(min_lr - i * (val_step/10))
    boundaries.append(cycle_epochs + i + 1)
boundaries.pop()

# Cut values if we start training not from the first epoch
values_start_idx = 11
values = values[values_start_idx:]
print(values)

boundaries = boundaries[:-values_start_idx]
boundaries = [i * steps_per_epoch for i in boundaries]
print(boundaries)

```

```

epochs = cycle_epochs + extra_epochs - values_start_idx
print(epochs)

lr_schedule =
keras.optimizers.schedules.PiecewiseConstantDecay(boundaries, values)
# loss_fn = ssim_loss_fn
optimizer = keras.optimizers.Adam(learning_rate=lr_schedule)

```

Безпосередньо навчання:

```

model.compile(
    optimizer=optimizer, loss=loss_fn,
)

model.fit(
    train_ds, epochs=epochs, callbacks=callbacks,
    validation_data=valid_ds, verbose=2
)

```

Демонстрація роботи нейромережі:

```

total_bicubic_psnr = 0.0
total_test_psnr = 0.0

upscale_factor = 2

# for index, test_img_path in enumerate(test_img_paths[50:60]):
for index, test_img_path in enumerate(test_img_paths):
    img = load_img(test_img_path)
    lowres_input = get_lowres_image(img, upscale_factor)
    prediction = upscale_image(model, lowres_input)

    w = lowres_input.size[0] * upscale_factor
    h = lowres_input.size[1] * upscale_factor

    predict_img_arr = img_to_array(prediction)

    lowres_img = lowres_input.resize((w, h))
    lowres_img_arr = img_to_array(lowres_img)

    # lowres_img_arr = process_target(lowres_img_arr)
    # highres_img_arr = img_to_array(highres_img)
    highres_img_arr = tf.image.resize(img_to_array(img), (h,w))
    # highres_img_arr = process_target(highres_img_arr)
    # predict_img_arr = prediction
    bicubic_psnr = tf.image.psnr(lowres_img_arr, highres_img_arr,
max_val=255)
    test_psnr = tf.image.psnr(predict_img_arr, highres_img_arr,
max_val=255)

```

```

total_bicubic_psnr += bicubic_psnr
total_test_psnr += test_psnr

print(
    "PSNR of low resolution image and high resolution image is
    %.4f" % bicubic_psnr
)
print("PSNR of predict and high resolution is %.4f" % test_psnr)
# plot_results(lowres_img, index, "lowres")
# plot_results(img, index, "highres")
# plot_results(prediction, index, "prediction")

# print("Avg. PSNR of lowres images is %.4f" % (total_bicubic_psnr /
# 10))
# print("Avg. PSNR of reconstructions is %.4f" % (total_test_psnr /
# 10))

print("Avg. PSNR of lowres images is %.4f" % (total_bicubic_psnr /
len(test_img_paths)))
print("Avg. PSNR of reconstructions is %.4f" % (total_test_psnr /
len(test_img_paths)))

```